BIGMACHINES

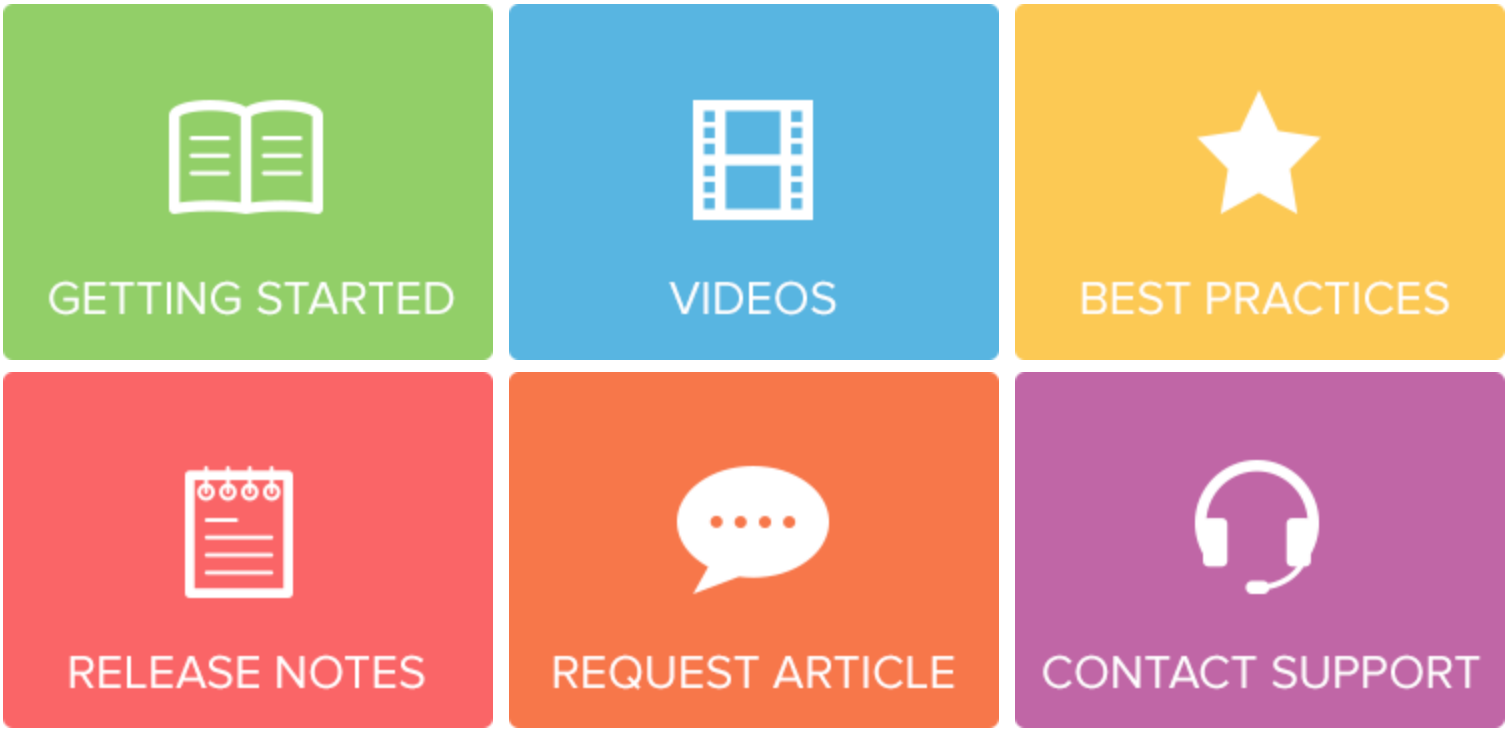
BIGMACHINES 2013
AUTUMN



12/20/2013

BigMachines 2013 Autumn
Product Documentation

The BigMachines Product Documentation is a consolidation of context-sensitive online help content and Release Notes for the BigMachines product.
© Copyright 2000-2013 BigMachines, Inc.


GETTING STARTED
VIDEOS
BEST PRACTICES
RELEASE NOTES
REQUEST ARTICLE
CONTACT SUPPORT

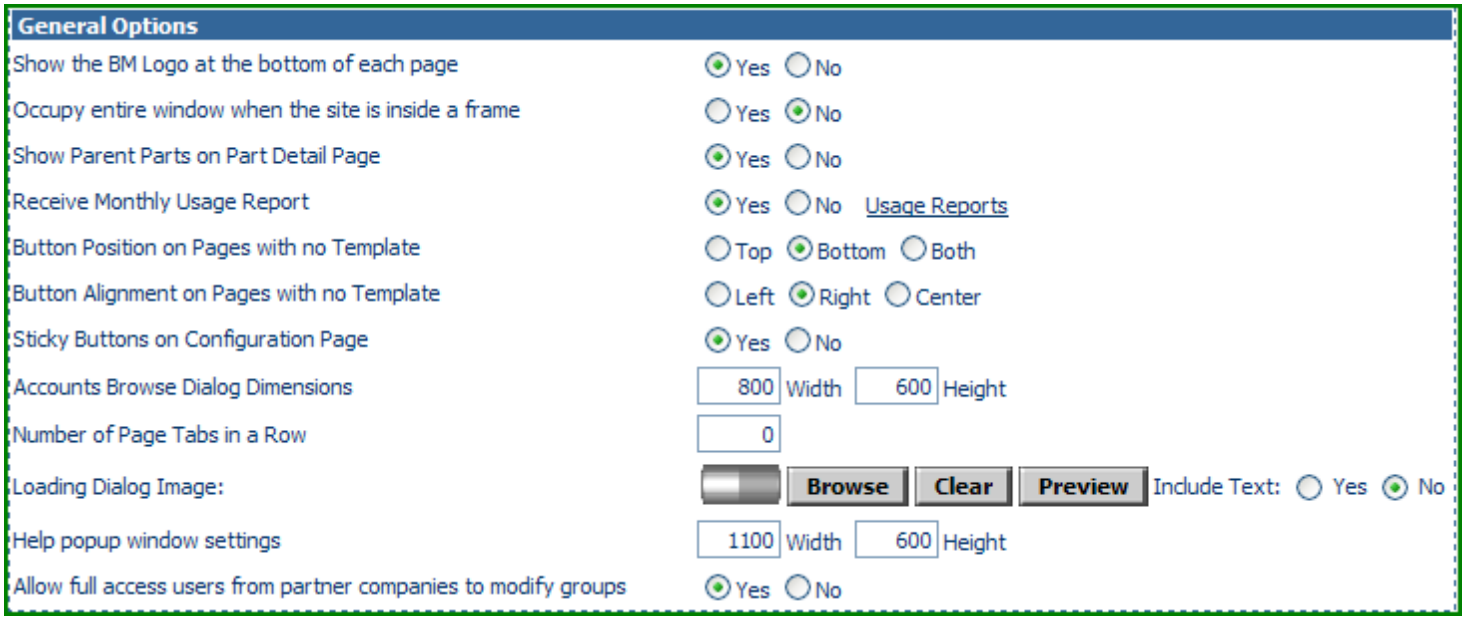
MANAGING GENERAL OPTIONS

OVERVIEW

You can specify general site options and system options in the areas of commerce and configuration. You can also access the File Manager to store files that are used system-wide. You can customize your Favorites List. You can also administer the set of Countries and/or State and set up validations based on the country of the user.

Feature	Description
Exchange Rates	Clicking Exchange Rates displays the Currency Conversion Rate List page using which you can manage exchange rates.
Unit Conversions	Clicking Unit Conversions displays the Conversion Class List page using which you can manage unit conversions.
Row Count per Page	Clicking Row Count Per Page displays the Iterator Administration page for defining iterator lengths.
Search Behavior	Clicking Search Behavior displays the Define Search Behaviors page for defining search behavior.
Usage Reports	Clicking on this link displays the Job Scheduler page for the Schedule Usage Reports.

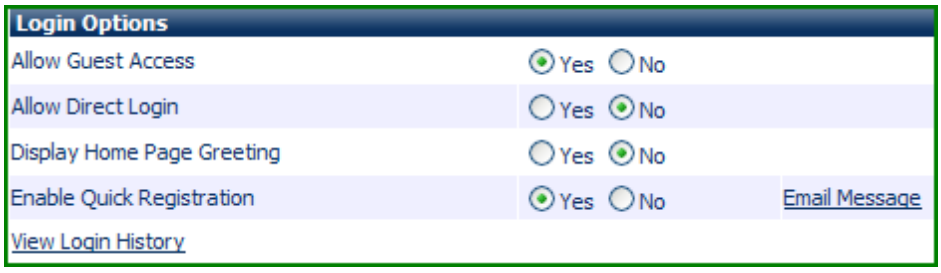
Setting up the General Options



1. Specify whether or not you want to **Show the BM Logo at the bottom of each page** by clicking **Yes** or **No**.
2. Specify whether or not you want to **Occupy entire window when the site is inside a frame** by clicking **Yes** or **No**.
3. Specify whether or not you want to **Show Parent Parts on Part Detail Page** by clicking **Yes** or **No**.
4. Determine if you would like to receive **Monthly Usage Reports** by clicking **Yes** or **No**.
5. Specify how you want **Button Position on Pages with no Template** to appear by clicking **Top**, **Bottom**, or **Both**.
6. Specify how you want **Button Alignment on Pages with no Template** to appear by clicking **Top**, **Bottom**, or **Both**.

7. Specify if you'd like to have **Sticky Buttons on the Configuration page**. *The sticky bar will follow the users as they scroll up and down.*
8. Specify the width and height for the **Accounts Browse Dialog Dimensions**.
9. Specify the **Number of Page Tabs in a Row**.
10. Specify a **Loading Dialog Image** by clicking **Browse** to search for and select a new logo.
    - Clicking **Browse** causes a dialog labeled File Manager - Browse to appear.
        - In this window, you can select a folder from which you want to import the new logo using the Folder drop-down list.
        - Selecting a folder name displays the logo images present in the folder, along with a Preview, Description, and the date and time the logo was imported to the folder.
        - Click Select the entry for the desired logo. *This closes the File Manager - Browse window, and displays the name of the selected image in the Home Page Set Up page.*
        - Specify whether or not you want to Include Text on the image by clicking **Yes** or **No**.
    - Click **Clear** to remove the selected image.
    - Click **Preview** to view the image selected.
11. Specify the Width and Height Dimensions for the Help Popup Window.
12. Specify if the Full Access User from Partner Organization is able to Modify Groups.
    - **If Yes:** then the Groups tab is visible to the Partner Organization Full Access Users in the User Profile.
    - **If No:** then the Groups tab is hidden and the Partner Organization Full Access User is unable to change the group of a user.

Setting up the Login Options



1. Specify whether or not you want to **Allow Guest Access** by clicking **Yes** or **No**. *Guest access permits visitors to view your online catalog of products and parts without registering on your site. You must have "YES" selected in order to use eCommerce Registration functionality.*

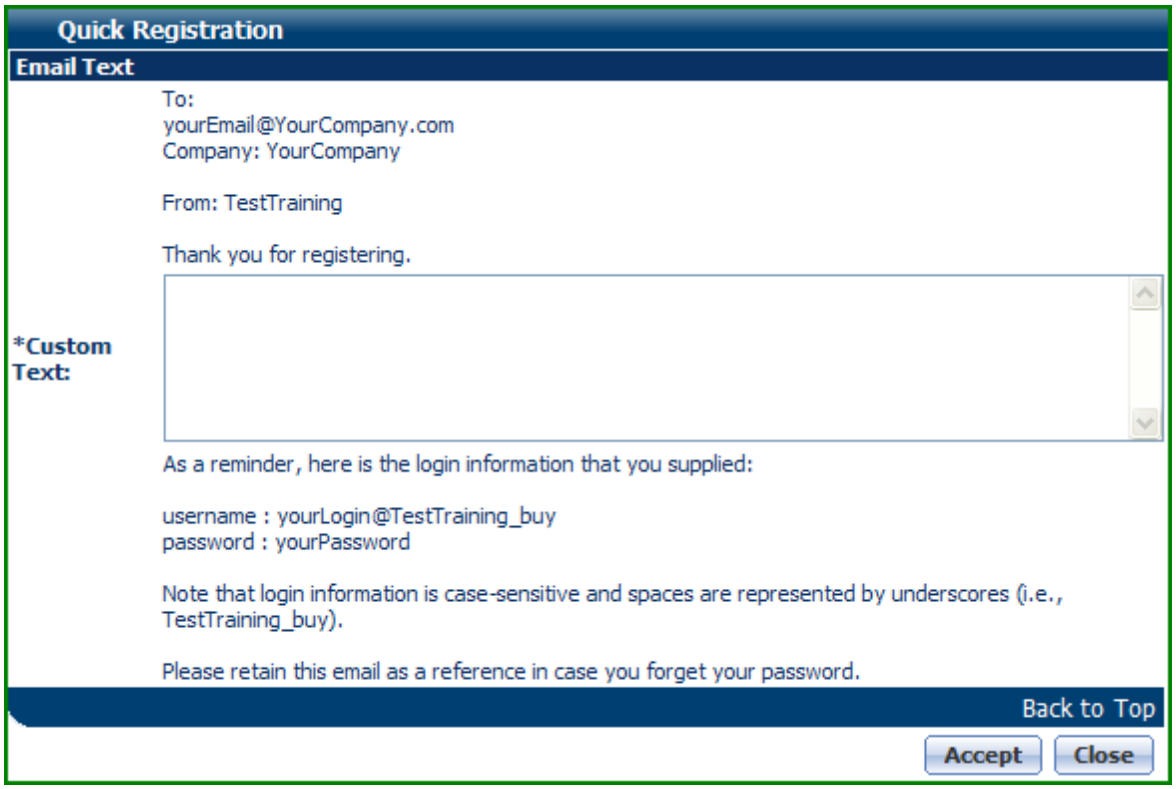
2. Specify whether or not you want to **Allow Direct Login** by clicking **Yes** or **No**. *Allowing Direct Logins gives third party (non-partner) access to your application without requiring a separate login.*

3. Specify whether or not you want to **Display Home Page Greeting** by clicking **Yes** or **No**.

4. Specify whether or not you want to **Enable Quick Registration** by clicking **Yes** or **No**. When quick registration is disabled, the system prevents users from registering for quick registration user accounts. Those users who already have accounts will still be able to log in. *You can grant visitors both guest access and quick registration access.*

    - Quick registration enables people visiting your site to easily create and sign up for a user account, and then log in with the account to shop and possibly make purchases. Quick registration users can only submit commerce documents if commerce processes permit them to do so.

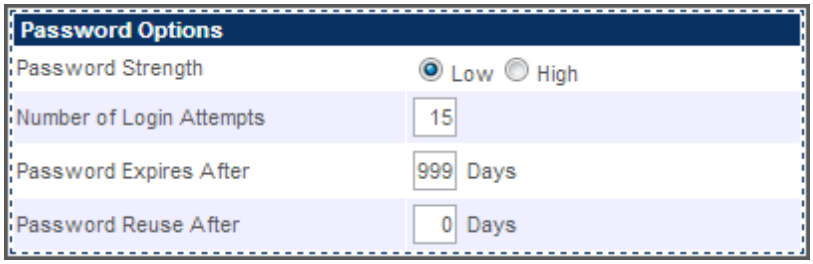
    - Clicking **Email Message** causes a dialog to open where you can customize the text that is sent to users after they submit a quick registration form. Click **Accept** to save changes made to the page.



5. Click **View Login History** to view the login history details for all users. *You may only view this page if you have the access rights.*

6. Notice that access changes take effect immediately after clicking **Accept**.

Setting up the Password Options



1. Navigate from: Admin Home Page > General Site Options > Password Options.
2. Specify the **Password Strength** as **Low** or **High**. *This is for providing greater security to your access.*
    - **High:** Requires 8-31 characters: at least one uppercase letter, at least one number, and at least one special character is REQUIRED.
    - **Low:** Requires 4-30 characters without special requirements.
3. Specify the **Number of Login Attempts** after which the user will be unable to attempt logging in.
4. Specify the number of days in which the **Password Expires After**. *This has to be a positive integer between 1-999. This field cannot be left blank.*
5. Specify the number of days for the **Password Reuse After**. *This specifies the number of days after which a password that has been used before and has expired can be reused.*

Legacy users can continue to use their old password of length less than 8 characters until they try to update their password. This change does not impact Forget Password functionality on the Unplugged client.

In compliance with ISO 27001, all BigMachines passwords must have an expiry. Once a site is upgraded to 2013 Spring, the maximum number of days allowed before a password expires is 999. *It is considered best practice to set the password expiration period to <90 days.*
-The password expiration time period begins once the site is upgraded.
-If the system determines that a user's last password update has exceeded the upgraded password expiration setting and the value was not previously set to expire, the user will be required to reset their password the next time they log in.

ADMINISTRATION

Modifying Email Signatures

This page displays the existing email signature that can be modified using the text field. The email signature that is defined and displayed in this page is attached to all emails sent by the system. **To create/edit a signature:**

1. Navigate from: Admin Home Page > General Email Signature.
2. Create a new/modify the current signature (using HTML) in the text field provided.
3. Click **Update**.

Defining Search Behavior

Use this page to set how you want to support wildcards for each search type. Wildcards are designated with an asterisk (*) and you can select one of the following options for wildcard support from the drop-down list:

- **Explicit** makes wildcards optional. You can enter them when needed.
- **Implicit** means the system automatically appends a wildcard to the search criteria.
- **Not Allowed** turns off wildcard support.

You can modify and the Leading and Trailing Wildcard support for the following search pages:

- Part Number Search Behavior
- Part Description Search Behavior
- Serial Number Search Behavior
- Accounts Company Search Behavior
- Accounts Customer ID Search Behavior
- Commerce Search Behavior
- Bulk Download Filter Behavior
- Data Table Name Search Behavior

After you have made the desired settings, click **Update**. *The changes will be reflected the next time when you visit a modified search page.*

Defining Interior Lengths

You can set different iterator lengths for each of the pages shown. The smaller the iterator, the faster a page will load. Use this page to make and modify the result set sizes for the following pages:

1. Segment List Page Result Set Sizes
    - **Product Line List Pages:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.

- **Model List Pages:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.



The default values are 20.

2. Part List Page Result Set Sizes
    - **Part Search Results Pages:** Select a value between 2-200 from the drop-down list to specify the result set size for the page.
    - **Serial Number Results Pages:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.
    - **Part Display Page Parent Part Results:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.



The default value for **Part Search Results Pages** and **Serial Number Results Pages** is 30.

The default value for **Part Display Page Parent Part Results** is 10.

3. Commerce Section Result Set Sizes
    - **Commerce List Page:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.
    - **Accounts Browse Dialog:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.



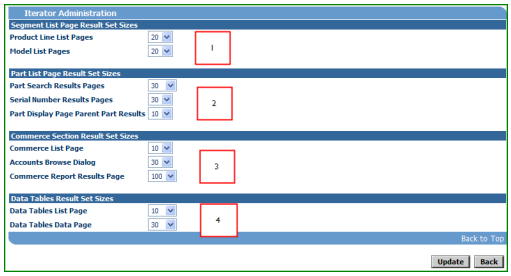
The default values are 50.

4. Data Tables Result Set Sizes
    - **Data Tables List Page:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.
    - **Data Tables Data Page:** Select a value between 2-200 from the drop-down list to specify the result set size for the

page.



The default values are 50.



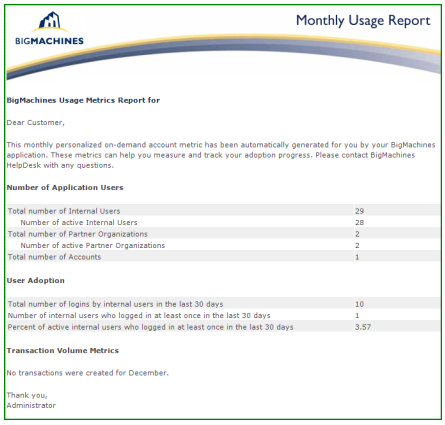
Scheduling Usage Reports

Site Metrics	Description
Number of Application Users	This section includes the Total Number of Internal Users, Number of Active Internal Users, Total Number of Partner Organizations, Total Number of Active Partner Organization and Total Number of Accounts.
User Adoption	This section includes the Total Number of logins by Internal Users in the last 30 days and the Number of internal users who logged in at least once in the last 30 days.
Transaction Volume Metrics	This section of the report includes the number of transactions created per Commerce Process by Internal and External users in the time period specified. It also includes the Total Number of Transactions created per commerce process by internal and external users till date. It also lists the number of transactions created per Internal User in the time period specified.

You can select a usage report for a particular month, selected from the last 6 months, to send to an email address in the **Usage Metrics** section.

1. Enter a mail ID of the person you need to mail the on-demand report to. *By default, this field is populated with the email address of the superuser; the Monthly Usage Report is sent to the company email address.* For more information, see Host Company.
2. Under **Report Duration**, select either of the following options:
   - **Month:** Select an appropriate month from the drop down menu.
   - **Start Date/End Date:** Enter the start date and end date of the scheduled job.
3. Click **Execute Now** to schedule the report.

Sample Usage Report:



Monthly Usage Report

BIGMACHINES

BigMachines Usage Metrics Report for

Dear Customer,

This monthly personalized on-demand account metric has been automatically generated for you by your BigMachines application. These metrics can help you measure and track your adoption progress. Please contact BigMachines HelpDesk with any questions.

Number of Application Users

Total number of Internal Users	29
Number of active Internal Users	28
Total number of Partner Organizations	2
Number of active Partner Organizations	2
Total number of Accounts	1

User Adoption

Total number of logins by internal users in the last 30 days	10
Number of internal users who logged in at least once in the last 30 days	1
Percent of active internal users who logged in at least once in the last 30 days	3.57

Transaction Volume Metrics

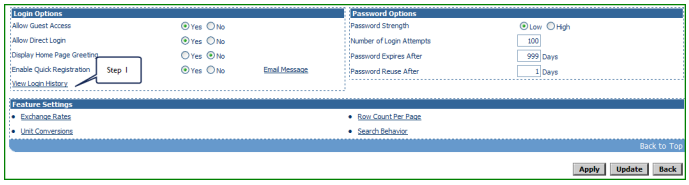
No transactions were created for December.

Thank you,
Administrator

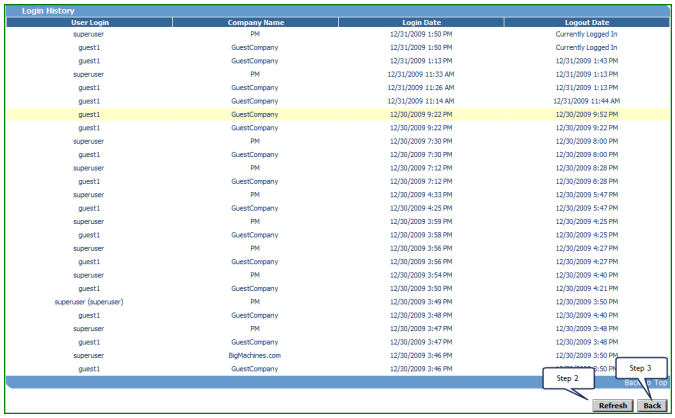
Viewing Login History

The User Login History page displays the login history information for the last 30 days from current date. *Proxy logins are indicated in parentheses.*

1. Click **View Login History**. *You must have access rights to view this page.* **Example:** if you are the superuser, you can view the login history details of all users.
    - **User Login:** The name of the user who logged in.
    - **Company Name:** The company name of the user.
    - **Login Date:** The login date and time.
    - **Logout Date:** The date and time the user logged out of the system.



2. Click **Refresh** to refresh the contents of the page.
3. Click **Back** to return to Managing General Options.



RELATED TOPICS

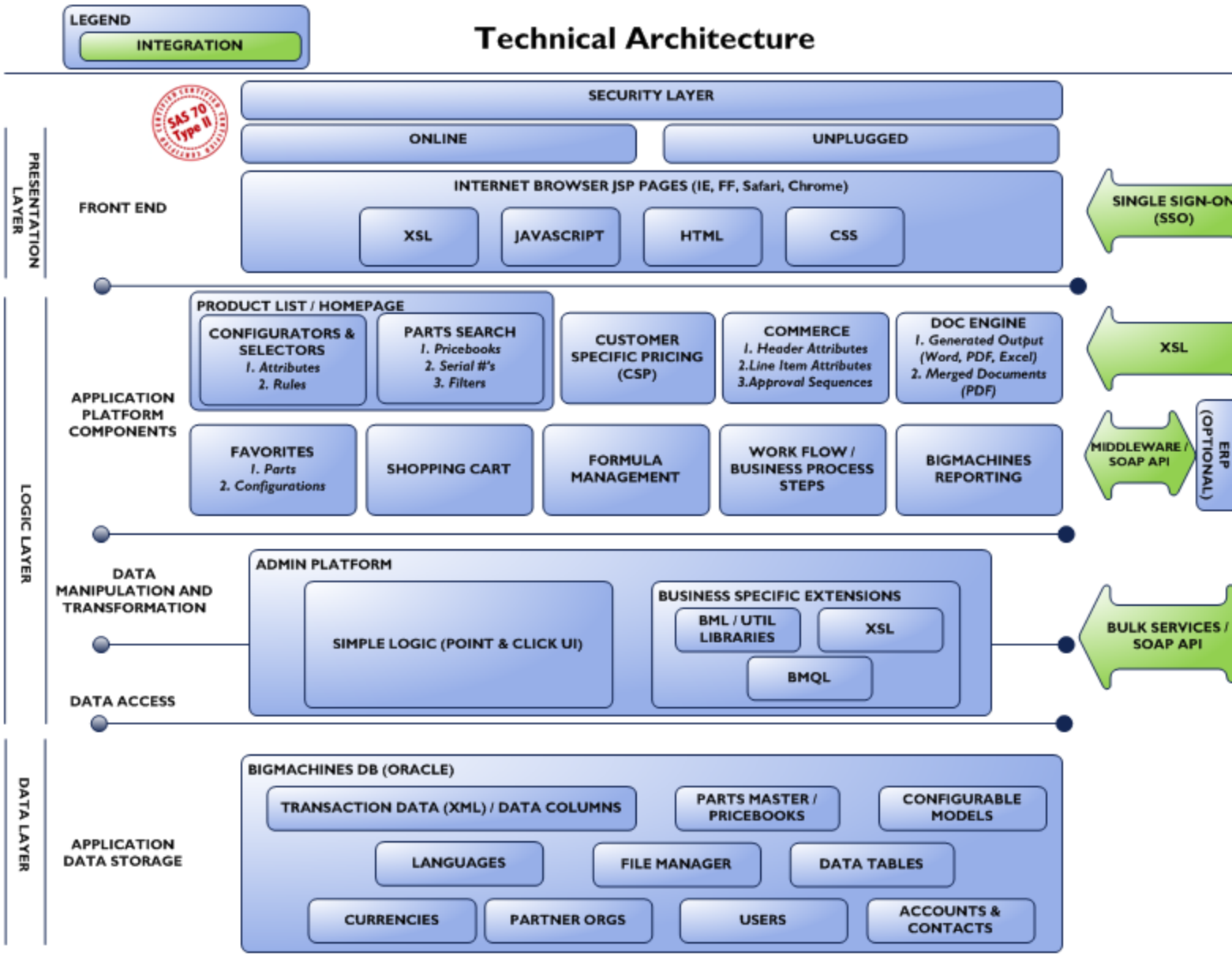
UNDERSTANDING BIGMACHINES

OVERVIEW

BigMachines digitizes selling processes and captures an organization's tribal knowledge to provide online product selection, configuration, quoting, and ordering capabilities for new products and aftermarket parts and services. We ensure effective quote and order management and provide powerful reports, analysis, and forecasts across all sales channels and customers. The application easily integrates to CRM and ERP systems to eliminate duplicate data entry and maintenance and to fully leverage the existing IT infrastructure.

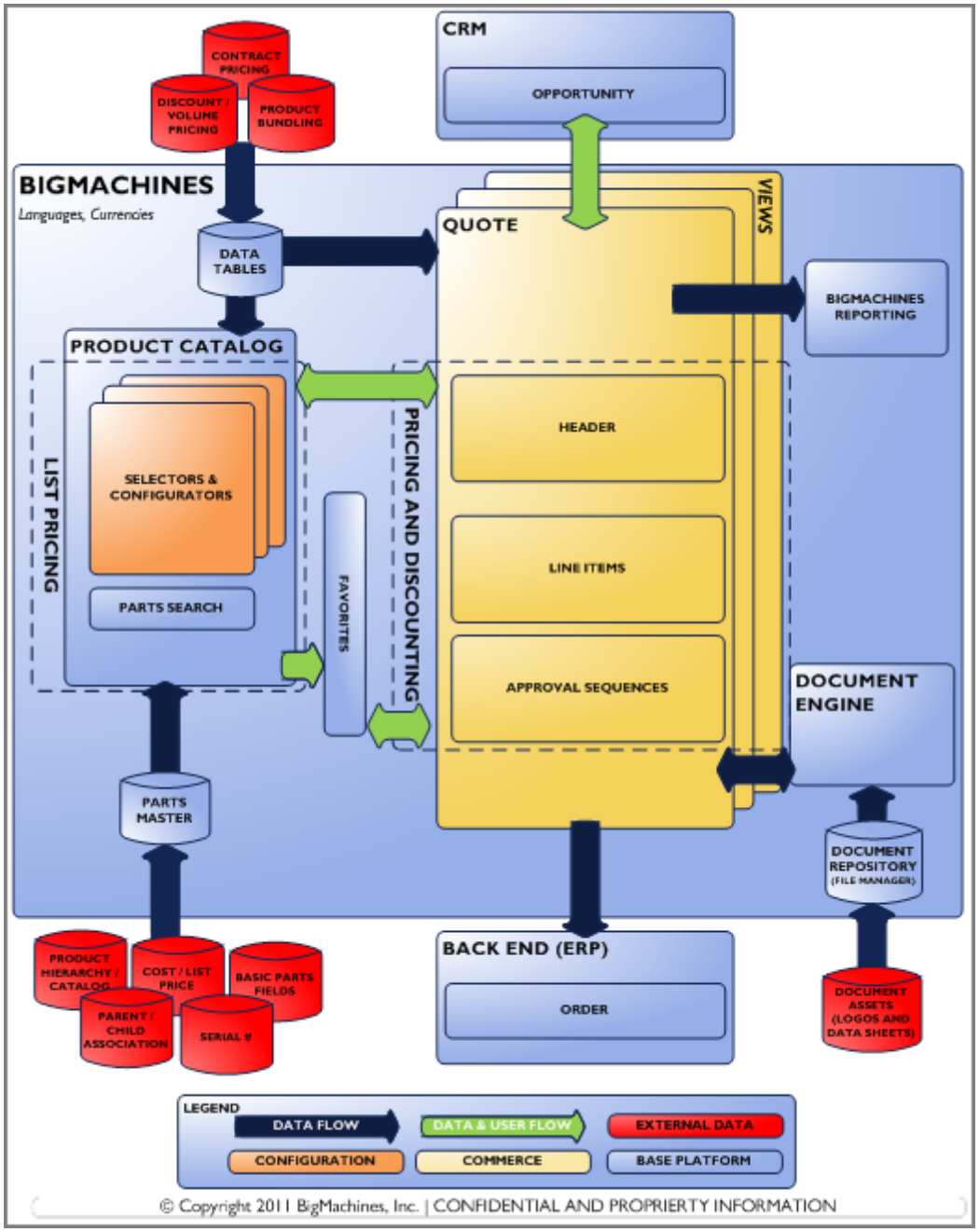
The BigMachines platform has been built for rapid, on-demand deployment through an intuitive web interface. Both users and system fullaccessusers only need a browser and a secure log-in to access BigMachines solution from anywhere in the world. As a result, our customer can rapidly implement and deploy our BigMachines solution globally.

Technical Architecture



© Copyright 2013 BigMachines, Inc. | CONFIDENTIAL AND PROPRIETARY INFORMATION

Solution Map



ADMINISTRATION

Catalog Page Reference

Page Name	Description
Part Search Results	Displays a list of parts, after a user has conducted a parts search.

Page Name	Description
Part Display	Displays part details. This page appears if a parts search results in a single return or if a user clicks a part number from the Parts Search Results page.
Part Associations	Displays sub-parts associated with a part. This page appears after a user clicks on a part listed on the Part Display page.
Product Line List	Displays a list of product lines, after a user has selected a product family or conducted a product line search.
Product Line Profile	Displays product line details for a single product line. This page appears if a user selects a product line from the Product Line list page. Information here comes from the Product Line description.
Model Profile	Displays model details for a single model. This page appears if a user clicks a model from the Model List page.
Model Configuration	Displays configurable attributes for configurable models - what is typically referred to as Configuration. This page is accessible when a model has configurable attributes only. Process invocation buttons created for this page typically display on the last page of the configuration flow.
Favorites List	Displays and maintains a list of products selected as Favorites. Users can add products to the Favorites list and then make purchases from the list.
Product Line Compare	Displays a side-by-side comparison of product lines. Users can access this page from the Product Line List page, after selected a set of product lines to compare.
Recommended Items	Displays recommended items/parts available for a single model. This is shows on the model configuration page, if recommended item rules are used to recommend mandatory or optional items for a configuration.
Bill of Materials for Serial Numbers	Displays a BOM after a user clicks a serial number on the Serial Number Search Results page.
Multiple Parts Buy	Allows users to submit quotes, orders or purchases for more than one part. This page is available when the Part Search Template is used on the commerce home page.

Viewing Version Details

Navigation: Admin Home Page > Administration Platform header bar > Click "View BigMachines Version"

This page gives you basic set up information about your site, including the application version, major features in the release, partner integration, if any and memory statistics. **The following details can be viewed:**

- **Version:** The version of currently in use.
- **Release Date:** The date and time when the version was released.
- **Major Changes since the previous version:** This lists the new features available in this version.
- **Connection Pool Size:** The cache of database connections maintained by the database.
- **Database Driver Version:** The database version in use.
- **Run Task Thread:** If the task thread should be run or not.
- **WebMethods:** If this site is integrated with WebMethods.
- **Reporting:** If the reporting module is enabled.

- **Integration:** If this site is integrated with another service provider.
- **Configuration Cache:** If the configuration cache is enabled or not. Enabling the configuration cache, will make your configuration module faster.



Click **Refresh** to refresh the contents of the page. Click **Back** to go back to the main Administration page.

NOTES



Sites created with additional JavaScript, alternate CSS, or other custom functionality, may no longer be compatible with our supported browsers. Please contact Customer Support to determine if workarounds and minor fixes are available. Add-on work may be required to support customizations.*Functionality can not be guaranteed on any other versions. Before submitting a case, please verify that you and/or your users who have experienced a problem are using a supported browser and version.*

RELATED TOPICS

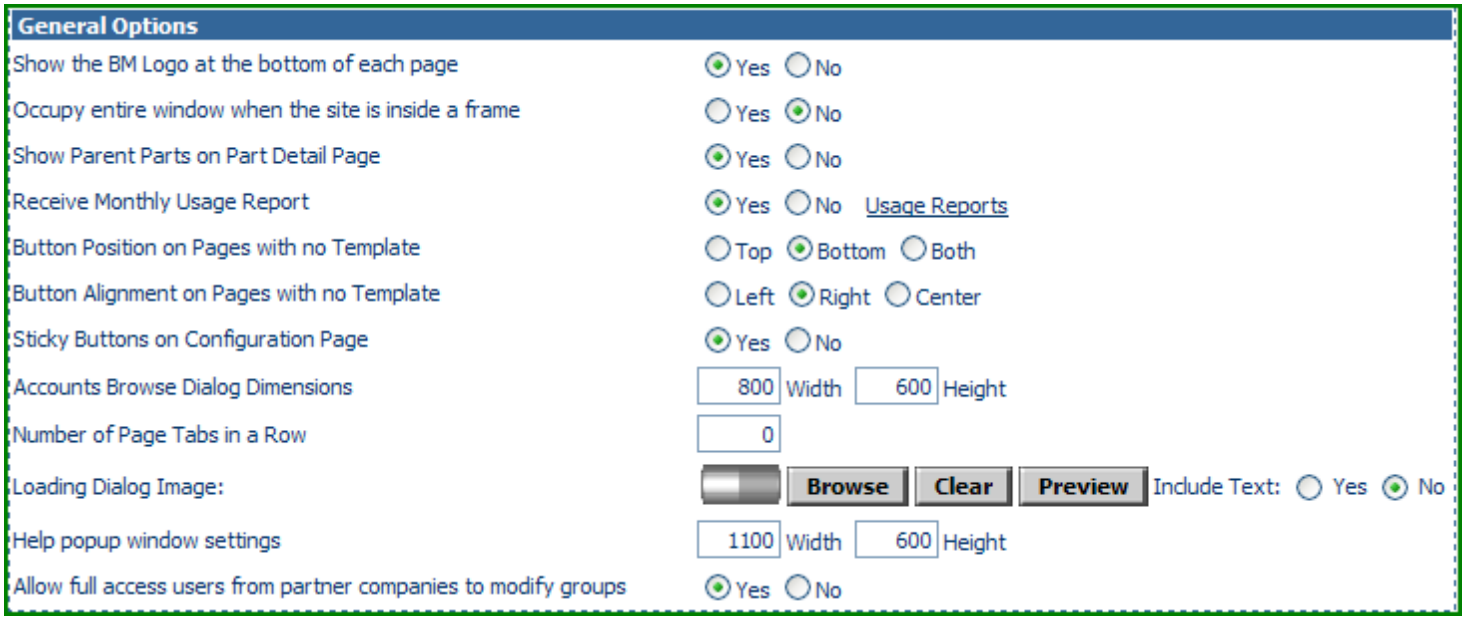
MANAGING GENERAL OPTIONS

OVERVIEW

You can specify general site options and system options in the areas of commerce and configuration. You can also access the File Manager to store files that are used system-wide. You can customize your Favorites List. You can also administer the set of Countries and/or State and set up validations based on the country of the user.

Feature	Description
Exchange Rates	Clicking Exchange Rates displays the Currency Conversion Rate List page using which you can manage exchange rates.
Unit Conversions	Clicking Unit Conversions displays the Conversion Class List page using which you can manage unit conversions.
Row Count per Page	Clicking Row Count Per Page displays the Iterator Administration page for defining iterator lengths.
Search Behavior	Clicking Search Behavior displays the Define Search Behaviors page for defining search behavior.
Usage Reports	Clicking on this link displays the Job Scheduler page for the Schedule Usage Reports.

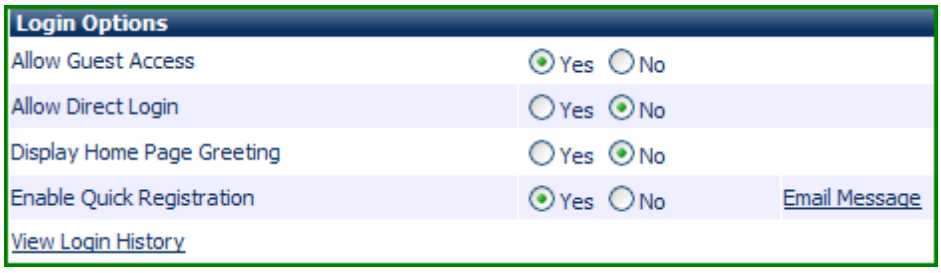
Setting up the General Options



1. Specify whether or not you want to **Show the BM Logo at the bottom of each page** by clicking **Yes** or **No**.
2. Specify whether or not you want to **Occupy entire window when the site is inside a frame** by clicking **Yes** or **No**.
3. Specify whether or not you want to **Show Parent Parts on Part Detail Page** by clicking **Yes** or **No**.
4. Determine if you would like to receive **Monthly Usage Reports** by clicking **Yes** or **No**.
5. Specify how you want **Button Position on Pages with no Template** to appear by clicking **Top**, **Bottom**, or **Both**.
6. Specify how you want **Button Alignment on Pages with no Template** to appear by clicking **Top**, **Bottom**, or **Both**.

7. Specify if you'd like to have **Sticky Buttons on the Configuration page**. *The sticky bar will follow the users as they scroll up and down.*
8. Specify the width and height for the **Accounts Browse Dialog Dimensions**.
9. Specify the **Number of Page Tabs in a Row**.
10. Specify a **Loading Dialog Image** by clicking **Browse** to search for and select a new logo.
    - Clicking **Browse** causes a dialog labeled File Manager - Browse to appear.
        - In this window, you can select a folder from which you want to import the new logo using the Folder drop-down list.
        - Selecting a folder name displays the logo images present in the folder, along with a Preview, Description, and the date and time the logo was imported to the folder.
        - Click Select the entry for the desired logo. *This closes the File Manager - Browse window, and displays the name of the selected image in the Home Page Set Up page.*
        - Specify whether or not you want to Include Text on the image by clicking **Yes** or **No**.
    - Click **Clear** to remove the selected image.
    - Click **Preview** to view the image selected.
11. Specify the Width and Height Dimensions for the Help Popup Window.
12. Specify if the Full Access User from Partner Organization is able to Modify Groups.
    - **If Yes:** then the Groups tab is visible to the Partner Organization Full Access Users in the User Profile.
    - **If No:** then the Groups tab is hidden and the Partner Organization Full Access User is unable to change the group of a user.

Setting up the Login Options



1. Specify whether or not you want to **Allow Guest Access** by clicking **Yes** or **No**. *Guest access permits visitors to view your online catalog of products and parts without registering on your site. You must have "YES" selected in order to use eCommerce Registration functionality.*

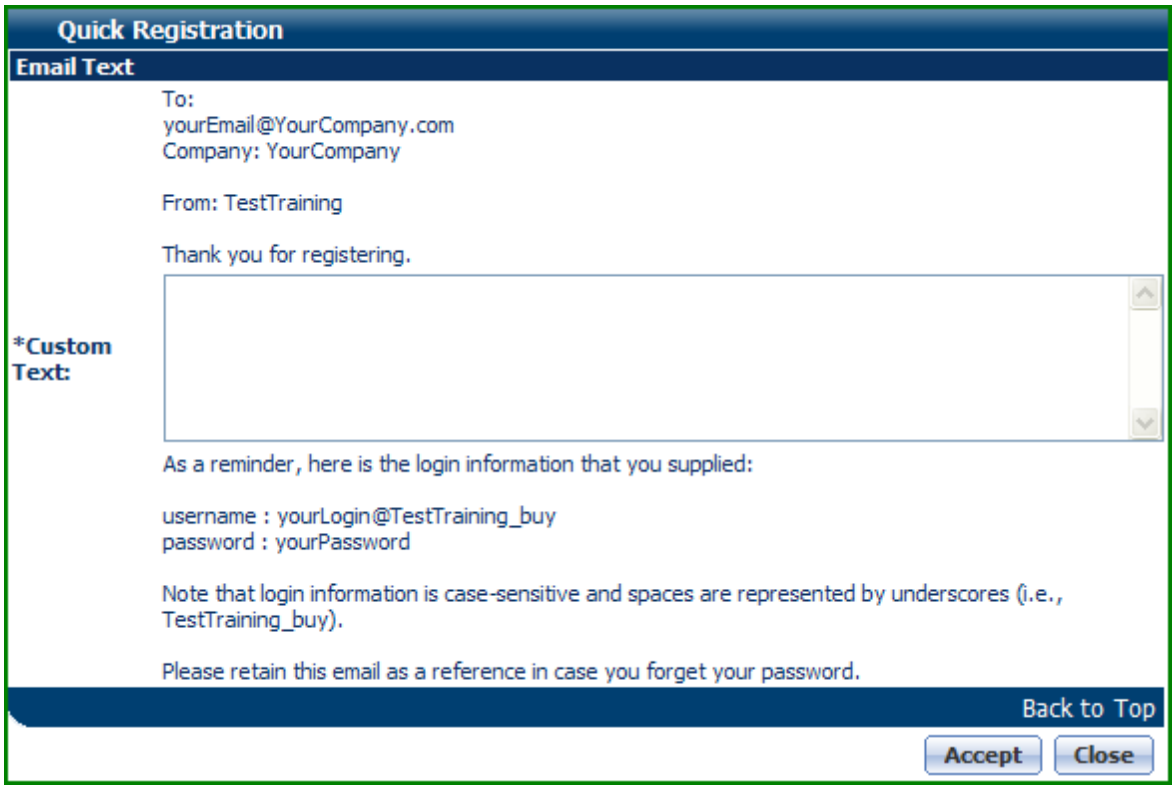
2. Specify whether or not you want to **Allow Direct Login** by clicking **Yes** or **No**. *Allowing Direct Logins gives third party (non-partner) access to your application without requiring a separate login.*

3. Specify whether or not you want to **Display Home Page Greeting** by clicking **Yes** or **No**.

4. Specify whether or not you want to **Enable Quick Registration** by clicking **Yes** or **No**. When quick registration is disabled, the system prevents users from registering for quick registration user accounts. Those users who already have accounts will still be able to log in. *You can grant visitors both guest access and quick registration access.*

    - Quick registration enables people visiting your site to easily create and sign up for a user account, and then log in with the account to shop and possibly make purchases. Quick registration users can only submit commerce documents if commerce processes permit them to do so.

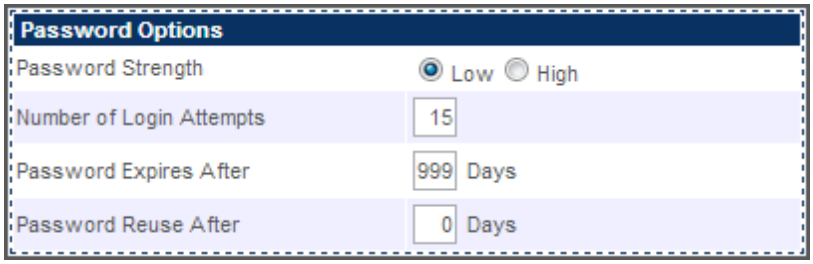
    - Clicking **Email Message** causes a dialog to open where you can customize the text that is sent to users after they submit a quick registration form. Click **Accept** to save changes made to the page.



5. Click **View Login History** to view the login history details for all users. *You may only view this page if you have the access rights.*

6. Notice that access changes take effect immediately after clicking **Accept**.

Setting up the Password Options



1. Navigate from: Admin Home Page > General Site Options > Password Options.
2. Specify the **Password Strength** as **Low** or **High**. *This is for providing greater security to your access.*
    - **High:** Requires 8-31 characters: at least one uppercase letter, at least one number, and at least one special character is REQUIRED.
    - **Low:** Requires 4-30 characters without special requirements.
3. Specify the **Number of Login Attempts** after which the user will be unable to attempt logging in.
4. Specify the number of days in which the **Password Expires After**. *This has to be a positive integer between 1-999. This field cannot be left blank.*
5. Specify the number of days for the **Password Reuse After**. *This specifies the number of days after which a password that has been used before and has expired can be reused.*

Legacy users can continue to use their old password of length less than 8 characters until they try to update their password. This change does not impact Forget Password functionality on the Unplugged client.

In compliance with ISO 27001, all BigMachines passwords must have an expiry. Once a site is upgraded to 2013 Spring, the maximum number of days allowed before a password expires is 999. *It is considered best practice to set the password expiration period to <90 days.*
-The password expiration time period begins once the site is upgraded.
-If the system determines that a user's last password update has exceeded the upgraded password expiration setting and the value was not previously set to expire, the user will be required to reset their password the next time they log in.

ADMINISTRATION

Modifying Email Signatures

This page displays the existing email signature that can be modified using the text field. The email signature that is defined and displayed in this page is attached to all emails sent by the system. **To create/edit a signature:**

1. Navigate from: Admin Home Page > General Email Signature.
2. Create a new/modify the current signature (using HTML) in the text field provided.
3. Click **Update**.

Defining Search Behavior

Use this page to set how you want to support wildcards for each search type. Wildcards are designated with an asterisk (*) and you can select one of the following options for wildcard support from the drop-down list:

- **Explicit** makes wildcards optional. You can enter them when needed.
- **Implicit** means the system automatically appends a wildcard to the search criteria.
- **Not Allowed** turns off wildcard support.

You can modify and the Leading and Trailing Wildcard support for the following search pages:

- Part Number Search Behavior
- Part Description Search Behavior
- Serial Number Search Behavior
- Accounts Company Search Behavior
- Accounts Customer ID Search Behavior
- Commerce Search Behavior
- Bulk Download Filter Behavior
- Data Table Name Search Behavior

After you have made the desired settings, click **Update**. *The changes will be reflected the next time when you visit a modified search page.*

Defining Interior Lengths

You can set different iterator lengths for each of the pages shown. The smaller the iterator, the faster a page will load. Use this page to make and modify the result set sizes for the following pages:

1. Segment List Page Result Set Sizes
    - **Product Line List Pages:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.

- **Model List Pages:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.



The default values are 20.

2. Part List Page Result Set Sizes
    - **Part Search Results Pages:** Select a value between 2-200 from the drop-down list to specify the result set size for the page.
    - **Serial Number Results Pages:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.
    - **Part Display Page Parent Part Results:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.



The default value for **Part Search Results Pages** and **Serial Number Results Pages** is 30.

The default value for **Part Display Page Parent Part Results** is 10.

3. Commerce Section Result Set Sizes
    - **Commerce List Page:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.
    - **Accounts Browse Dialog:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.



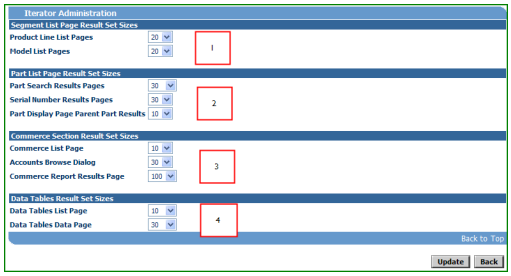
The default values are 50.

4. Data Tables Result Set Sizes
    - **Data Tables List Page:** Select a value between 2-50 from the drop-down list to specify the result set size for the page.
    - **Data Tables Data Page:** Select a value between 2-200 from the drop-down list to specify the result set size for the

page.



The default values are 50.



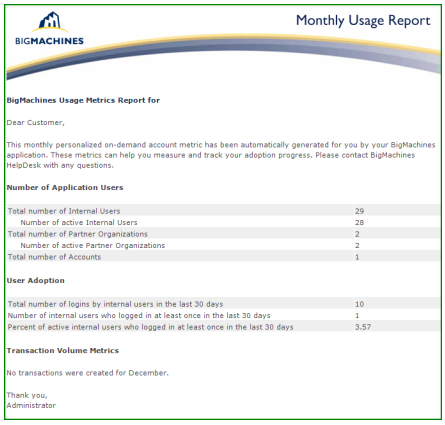
Scheduling Usage Reports

Site Metrics	Description
Number of Application Users	This section includes the Total Number of Internal Users, Number of Active Internal Users, Total Number of Partner Organizations, Total Number of Active Partner Organization and Total Number of Accounts.
User Adoption	This section includes the Total Number of logins by Internal Users in the last 30 days and the Number of internal users who logged in at least once in the last 30 days.
Transaction Volume Metrics	This section of the report includes the number of transactions created per Commerce Process by Internal and External users in the time period specified. It also includes the Total Number of Transactions created per commerce process by internal and external users till date. It also lists the number of transactions created per Internal User in the time period specified.

You can select a usage report for a particular month, selected from the last 6 months, to send to an email address in the **Usage Metrics** section.

1. Enter a mail ID of the person you need to mail the on-demand report to. *By default, this field is populated with the email address of the superuser; the Monthly Usage Report is sent to the company email address.* For more information, see Host Company.
2. Under **Report Duration**, select either of the following options:
   - **Month:** Select an appropriate month from the drop down menu.
   - **Start Date/End Date:** Enter the start date and end date of the scheduled job.
3. Click **Execute Now** to schedule the report.

Sample Usage Report:



Monthly Usage Report

BIGMACHINES

BigMachines Usage Metrics Report for

Dear Customer,

This monthly personalized on-demand account metric has been automatically generated for you by your BigMachines application. These metrics can help you measure and track your adoption progress. Please contact BigMachines HelpDesk with any questions.

Number of Application Users

Total number of Internal Users	29
Number of active Internal Users	28
Total number of Partner Organizations	2
Number of active Partner Organizations	2
Total number of Accounts	1

User Adoption

Total number of logins by internal users in the last 30 days	10
Number of internal users who logged in at least once in the last 30 days	1
Percent of active internal users who logged in at least once in the last 30 days	3.57

Transaction Volume Metrics

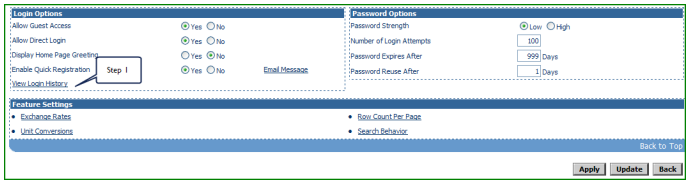
No transactions were created for December.

Thank you,
Administrator

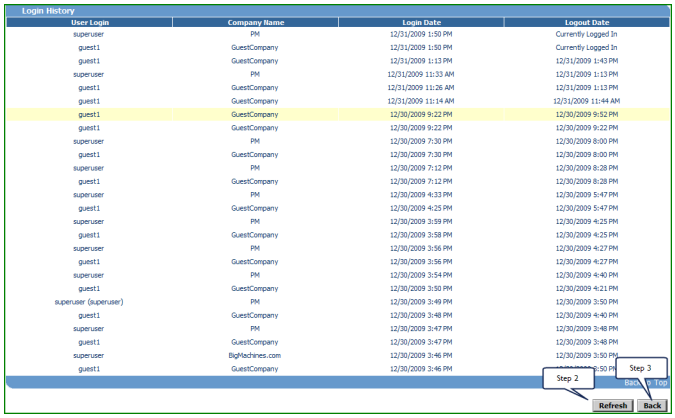
Viewing Login History

The User Login History page displays the login history information for the last 30 days from current date. *Proxy logins are indicated in parentheses.*

1. Click **View Login History**. *You must have access rights to view this page.* **Example:** if you are the superuser, you can view the login history details of all users.
    - **User Login:** The name of the user who logged in.
    - **Company Name:** The company name of the user.
    - **Login Date:** The login date and time.
    - **Logout Date:** The date and time the user logged out of the system.



2. Click **Refresh** to refresh the contents of the page.
3. Click **Back** to return to Managing General Options.



RELATED TOPICS

ADMINISTRATION HOME PAGE

OVERVIEW

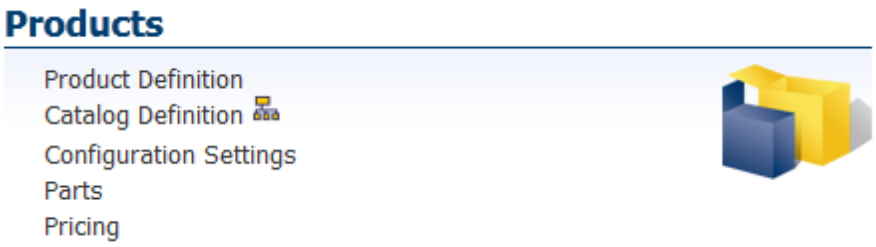
The BigMachines Administration Home Page is a rich, powerful tool where the BigMachines system fullaccessuser to access the application at any time. The administration platform (admin-side) allows you to set up your users, accounts, partner organizations, catalog, parts, configuration, and commerce processes. You can customize the look and feel of your system, including the colors, fonts, header and footer used in the system. Navigation flows and page templates can be customized to fit your needs. Administrative tools such as bulk upload/download, integration and File Manager make administration easier.

ADMINISTRATION

Major components of the BigMachines Admin Platform:

Products



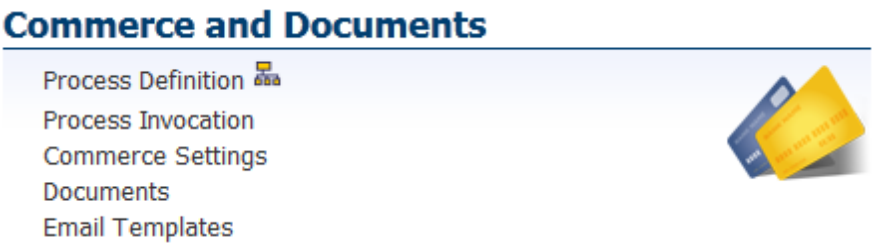


The Configuration section involves defining and configuring Products and Catalogs. This section allows you to manage your configuration models including the product hierarchy, attributes, and rules. Prior to creating your catalog items, you must first define the product structure for your vertical. Based on this structure, you can create your product hierarchy tree. Following this, you can set up your configurable attributes and rules to enable configuration.

- **Product Definition:** Define your product structure, i.e. define meta data definitions for your catalog items.
- **Catalog Definition:** Create your product hierarchy tree with configurable models, attributes and rules.
- **Configuration Settings:** Define configuration settings including recommended items and configurable pricing options.
- **Parts:** Parts data includes part numbers, serial numbers, and associations. For more information, see Parts - An Overview.
- **Pricing:** Define customer-specific pricing on a part by part basis.

Commerce and Documents





Here, you can set up your quoting, ordering, approval, and other workflow processes. By creating ordered sets of commerce documents, associated attributes and actions, you can enable buyers and supplier agents to conduct commerce on your system. Built from the ground up, each commerce process can be customized according to how you want to structure data flow between users during transactions. You can create one process for all commerce transactions or multiple processes based on different purchase types.

- **Process Definition:** Define your commerce documents, attributes, actions and proposal views. For more information, see Commerce Process Overview.
- **Process Innovation:** Define rules that enable the user to invoke your commerce process from various pages in the application including configuration and parts pages.
- **Commerce Settings:** Define options for commerce including toggling settings for allowing commerce processes to be cloned and migrated. Fore more information, see Commerce Process Settings.
- **Documents:** Create and manage documents within the Document Engine. For more information, see Document Engine Overview.
- **Email Templates:** Use a visual editor to create approval email templates. You can include main-document attributes and images. For more information, see Email Templates Overview.

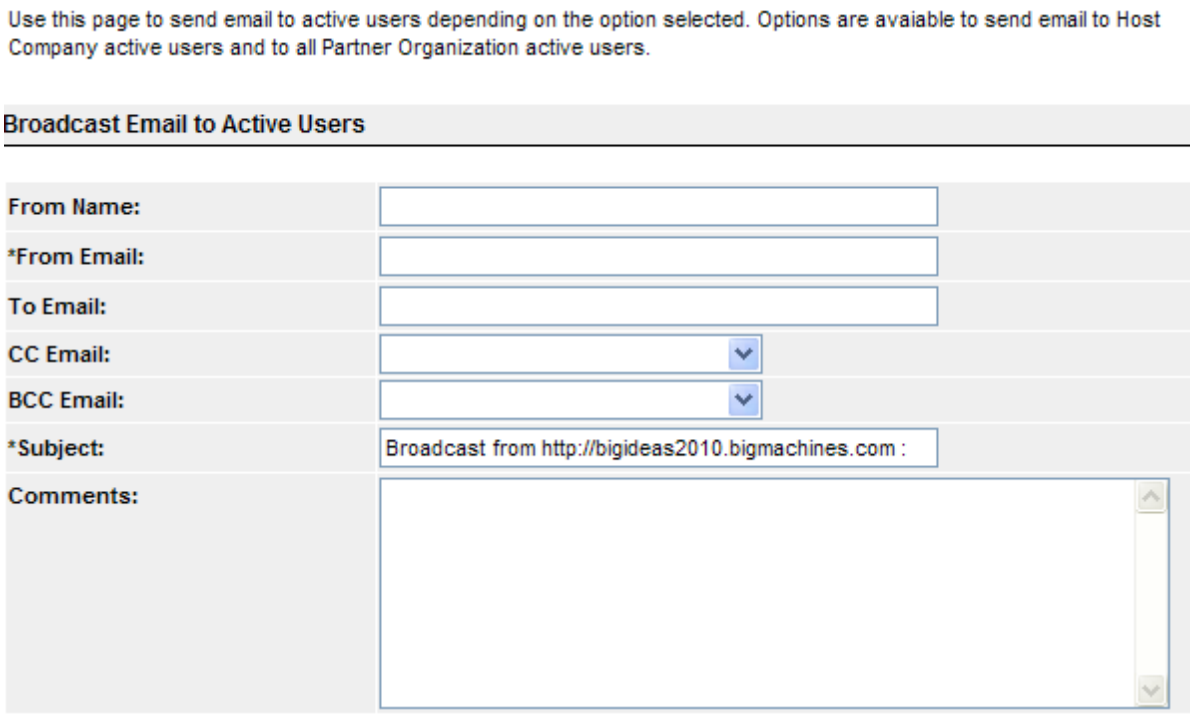
Utilities





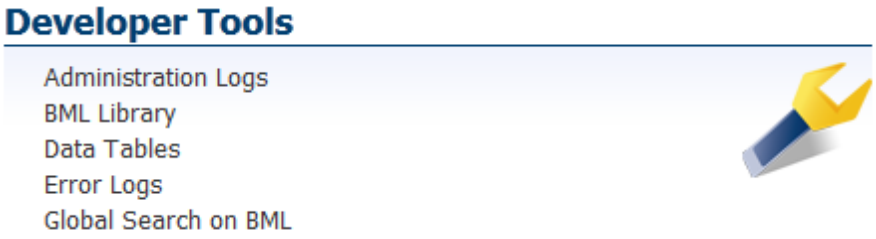
The BigMachines Administration Platform supports upload and download functions for the purpose of editing large amounts of system data. You can upload and download system data in bulk, including models, configurable attributes and rules, parts, account records, transactions, and commerce processes. You can also manage your company files and images through the File Manager. Text Administration will allow you to define custom text for your application.

- **Migration (Status):** Manage migrations between your test and production sites.
- **Bulk Data Services:** Upload and/or Download system data, like models, configuration rules, data tables and more. For more information, see Bulk Uploads and Bulk Downloads.
- **File Manager:** Upload and store files and images for use throughout the application. For more information, see File Manager.
- **Text Administration:** Define custom text for your application.
- **Broadcast Email:** Create and send an email blast to active Host Company and/or Partner users.



Developer Tools



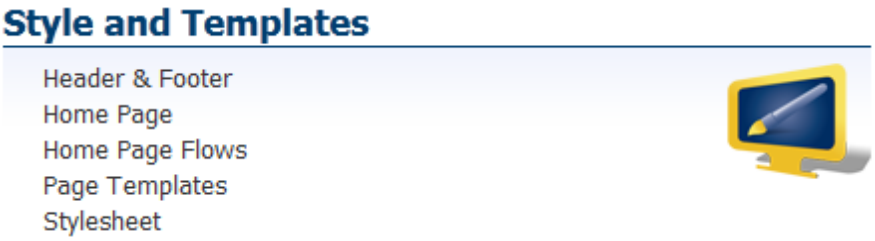


The data tables and BML sections help you build powerful custom solutions to meet your specific business use cases. These tools enhance the features already available in the administration platform.

- **Administration Logs:** Displays Log Messages for Admin changes done to Configuration and Commerce. Log includes user-name, product family, and date/time of the change. For more information, see Administration Log.
- **BML Library:** Create Util library functions to be used through Configuration and Commerce. For more information, see Library Functions.
- **Data Table:** Add data specific to your custom solution in data tables. You can also upload data through bulk upload or SOAP services. For more information, see Data Table Overview.
- **Error Logs:** View error logs generated by your application for quick debugging.
- **Global Search on BML:** Locate the scripts you have created easily using the search feature.

Styles and Templates





Use this section to define, manage, and customize your site branding and colors. You can define a custom HTML header, footer, and email signature here. One of the key features available in this section is the ability for the fullaccessuser to define your portal home

page. You can create a completely custom home page or use templates provided by BigMachines to quickly set up a standard home page.

- **Headers and Footers:** Define your custom header and footer. For example, adding the company logo to the application can be done in the header. For more information, see Header & Footer.
- **Home Page:** Define and deploy your application's home page in this area. For more information, see Home Page.
- **Home Page Flows:** Define home page flows to guide the user to pages in your application from your portal home page. *If you are using the standard home page, you can ignore this section.* For more information, see Home Page Flows.
- **Page Templates:** Define templates for different pages in your application. For example, recommended items section on the model configuration page. For more information, see Page Templates.
- **Stylesheet:** Define your site branding and colors using the main application stylesheet. You can even define custom stylesheets in this section. For more information, see Stylesheet Manager.

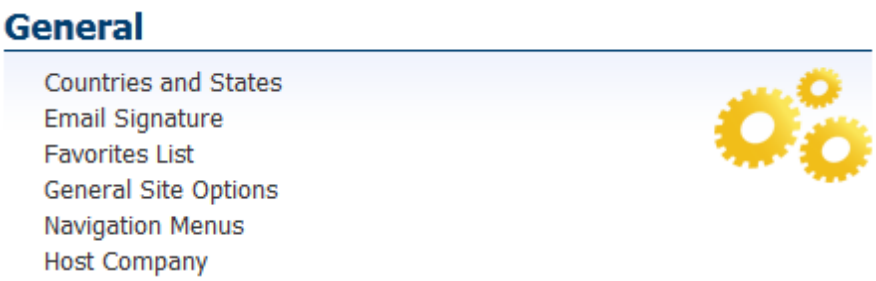
Text fullaccessusers:

It is possible to substitute default system text with words and phrases that match the vocabulary of your company. You can edit text like headings, navigation links, error messages, and any other system-generated text. Default text always remains in the system, so edited phrases can switch back to original system text at any time. This page displays the list of English alphabets. Clicking an alphabet displays list of words and phrases that can be modified in this administration area. Words and phrases are organized in alphabetical order (based on original text). **Using this page, you can:**

- Select a letter (Search String) to see all editable words starting with the letter selected.
- Select Others (Search String) to see all editable words starting with mathematical or other symbols that do not belong to the English Alphabets.
- View the current values for the Keys.
- Select a Key to display a new page for editing the selected text.
- While modifying the text, select the Override option before entering new content. In case Override is selected on the edit page, you can see an indicator in the Overridden column.

General



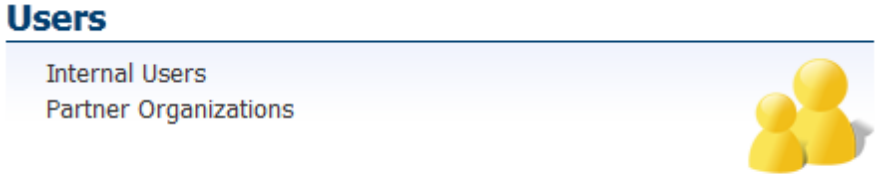


You can manage your site system setting ranging from the general setting, internationalization, and file manager. You can customize your application settings for configuration, commerce and/or your favorites. In addition, you can define and customize your application's state and country options. The global system settings also allows you to view error logs for quick debugging. You can also access the file manager from this section.

- **Countries and States:** Define states and countries options for your entire application including users and accounts.
- **Email Signature:** Define an email signature for any email being sent from within the application.
- **Favorites List:** Define options for updating your favorites for items to be reconfigured. For more information, see Favorites List.
- **General Site Options:** Define your general application settings including password options, login options and feature settings for exchange rates and unit conversions. For more information, see Managing General Options.
- **Navigation Menus:** Define the links to guide your users to different sections of your application.
- **Host Company:** Manage your company information, including supported currencies and languages. For more information, see Host Company.

Users



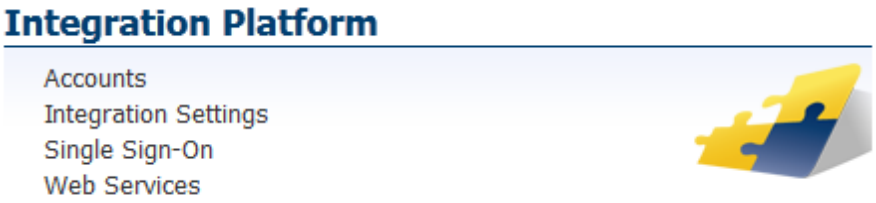


You can manage user access rights here; including partner organization access rights. You can manage your internal users, such as the ability to activate and inactivate them. Your application profile is maintained in the host company details that can be accessed from this section.

- **Internal Users:** Create, activate and inactive internal users from this section. For more information, see User Access Rights.
- **Partner Organizations:** Create, delete, and manage access rights to partner organizations from here. *You can also manage partner organization users from within the partner organizations*. For more information, see Partner Organizations.

Integration Platform





The BigMachines Administration Platform supports an integration layer that provides various types of partner integrations including back-end ERP systems and on-demand web applications. The platform also supports integration with LDAP servers to help manage your user accounts.

- **Accounts:** Create, edit and delete accounts information.
- **Integration Settings:** Define your communication parameters for real-time integration using middleware applications to connect to backend ERP systems. For more information, see Integration Guide.
- **Single Sign-On:** Manage authentication to automatically log into BigMachines without the need for multiple logins and re-authentication. For more information, see Single Sign-On (SSO).
- **Web Services:** Test section that helps you use the SOAP services API provided by the integration platform to integrate with on-demand web applications. For more information, see Web Services.

RELATED TOPICS

ADMINISTRATION LOG

OVERVIEW

Admin Change Logs maintain time stamped logs of administrative changes in commerce and configuration. All add, update, delete and deploy events in these modules are recorded with the details of the user that performed them. They are maintained for add, update, delete and deploy events in commerce and configuration only. By default the log of the current month is displayed in reverse chronological order so that the most recent is on the top and oldest one is at the bottom. The fullaccessuser can search for logs between a given period of time or by the last 6 months. After searching for logs, the result can also be downloaded in CSV format.

ADMINISTRATION

Searching for Administration Logs

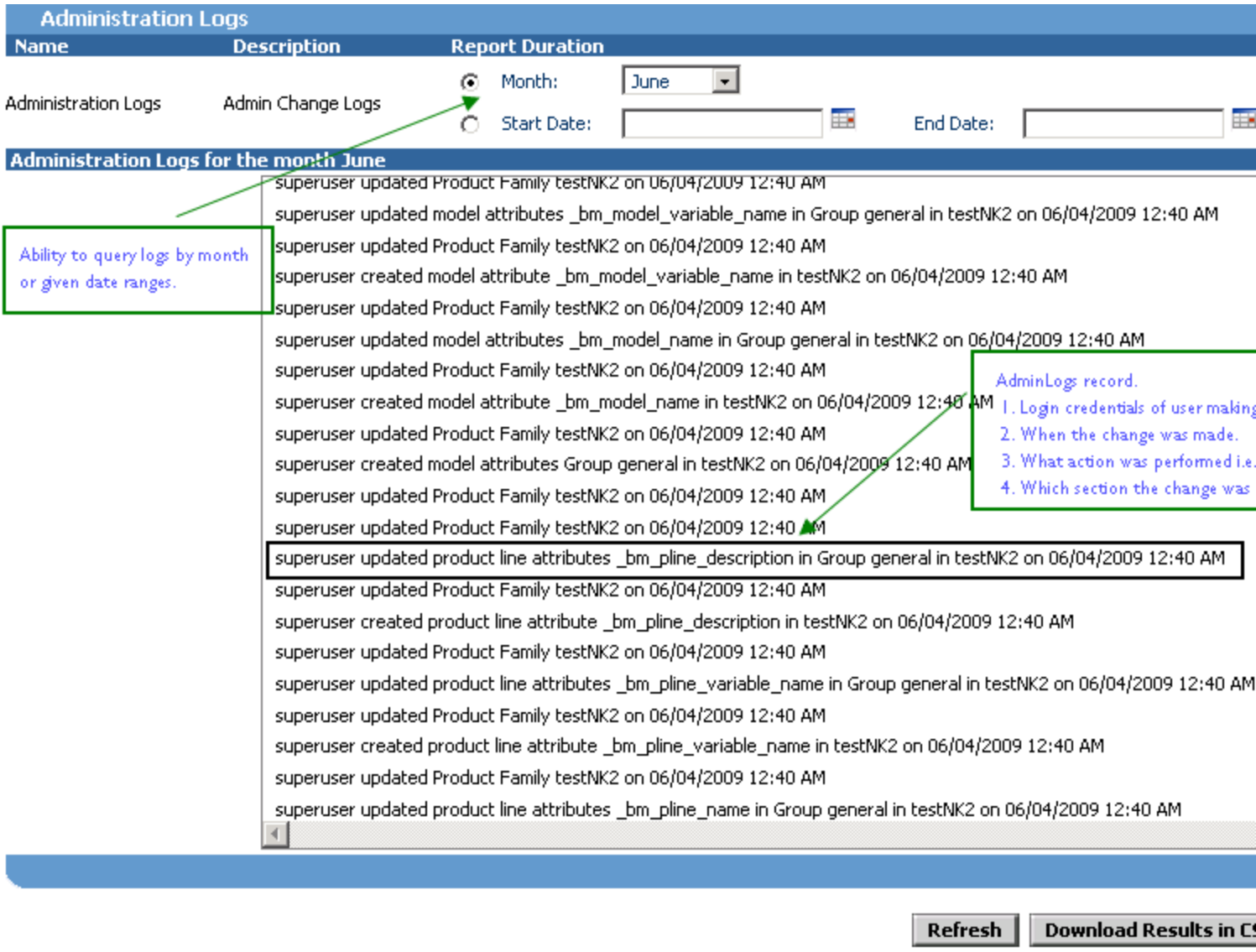
1. Select the **Month** for which the log has to be displayed from the drop-down.

   OR

   Specify the **Start Date** and **End Date**.

2. Click **Search** to search for logs within the specified time period.

3. View the logs.



Downloading Administration Logs

1. Click **Download Results in CSV Format**after searching for administration logs.

NOTES



All implicit actions are recorded in the logs when the fullaccessuser performs an add/update/delete or deploy type operation.
Example: if the fullaccessuser updates the Document Views tab of a commerce attribute, updates to the attribute and the implicit update to the workflow step are both recorded in the logs.

RELATED TOPICS

BULK DOWNLOADS

OVERVIEW

You can obtain and make changes to large quantities of system data here by downloading data in bulk, making batch edits, and then uploading it. Downloaded files are stored on the server. Files that are stored on the server can be downloaded onto your local system or viewed in your browser from the server location.

Downloadable Files

CSV	XML
Accounts (including Addresses)	Accounts (including Addresses)
Parts, Part & Pricebook Associations	Parts, Part & Pricebook Associations
Serial Numbers & Serial Number Associations	Serial Numbers & Serial Number Associations
User & Company Information	User & Company Information
Configurable Attributes	Configurable Attributes
Product Lines	Product Lines
Models	Models
File Manager	File Manager
Translations	Translations
Data Table Change Logs	Data Table Change Logs
	Commerce Transactions & Processes
	Configuration Rules
	Data Tables
	Catalog & Configuration Data

Common Filter Fields

Filter Field Names	Description
id	The BigMachines system generates a unique ID for each item it stores, be it a rule, attribute, account record, ... etc. For downloads, the ID field can be used to filter a single record in that download category. In general, you can locate an ID by navigating to the item you want to download and then viewing the corresponding ID in the URL.
company_id	Company IDs can be located in the home page URL.
ref_id	This is the ref_type ID that is generated (i.e.- Model ID, Product ID, Company ID for all Product Families). Navigate to the Model List Admin Page and view the ref_type ID located in the URL.
segment_id	For the "all product family" product category, the segment ID equals the company ID. Company IDs can be viewed on the commerce home page. For the "single segment" product category, the segment ID is a unique number associated with a particular segment. Segment IDs can be located by navigating to the Product Line Administration List page and viewing the segment_ID that displays in the URL.

Filter Field Names	Description
product_line_id	To locate a product line ID, navigate to the Model Listing Administration page and view the product_line_ID appearing in the URL.
model_id	To locate a model ID, navigate to a rule at the model level and view the model_ID appearing in the URL.
attribute_id	To located an Attribute ID, navigate to the Attribute Administration List page, putting the cursor over a specific attribute link and viewing the ID that dynamically appears in the browser window's bottom bar.
rule_id	To locate a Rule ID, navigate to the Rule List page, putting the cursor over a specific rule link and viewing the ID that dynamically appears in the browser window's bottom bar.
part_id	To locate a Part ID, navigate to the Part Administration page for a specific part and viewing the "pid" in the URL.
serial_number_id	To locate a serial number ID, navigate to the Serial Number Administration page for a specific serial number and viewing the "id" in the URL.
part_association_id	Part association IDs can be found on the Part Tree for Part Number page which displays part associations for a select part. To view a part association ID, highlight a part number and view the ID that dynamically appears in the browser window's bottom bar.
parent_part_id, child_part_id	When downloading a single part association, these IDs must be used together. Each field expects a part_id, one for the parent part association and the other for the child part association. Part associations can also be downloaded using the part_association_id instead of these two IDs.
serial_number_id, part_id	When downloading a single serial number association, these IDs must be used together. The first field expects the serial_number_id, and the second field expects the part ID for the associated part.

ADMINISTRATION

Managing Downloads

1. Select a category to download. *For Product Line, Model, and Catalog and Configuration Data, select a product family in addition to the category using the drop-down list provided.*
2. Click **Next** to go to the Search Criteria page.
3. Specify the criteria. *Only the data that satisfies the criteria will be downloaded.*
4. Click **Download**.



The types of configuration rules you can download are: recommendations, constraints and pricing.

Only one type of rule at a time can be downloaded. *Product line and model data can be downloaded for one segment at a time.*

Viewing Download Status

The Download Status page shows the module name and status of all files scheduled for download in the previous seven days. The page also displays the status message and the scheduled date and time.*Successful downloads showing a "completed" status are removed from the download status page after seven days. Download tasks with any status other than "completed" are not removed from the status page.* **Using this page, you can:**

- Sort the details based on Status, Status Message, and Scheduled Date/Time. *Click the appropriate heading to sort the details in the order you wish to view.*
- View the status of the task. *The task displays a "pending" status until the process begins. Finished downloads display a "completed" status.*
- Click a Module name that is to be downloaded. This opens a new pop-up window displaying the name of the .zip file. *You can select whether you want to Open the file using one of the drop-down options provided, or Save it to disk.*
- Remove local downloads from the server to free up room for future download files. Select all or a required Module that is to be deleted using the Select All or Select options, and click Delete to delete it. *Up to five download files can be stored on the server at any given time.*
- Select a task and click Stop Current Task to stop the download.
- Go back to the Page List page to setup and activate commerce invocations.



If the download file you want to view does not appear on the Downloaded XML Files page, the download process may not have completed.



Download tasks can have any of the following status: completed, failed, running, or pending. *If a download fails, retry the download process. If you cannot get download to succeed, contact an implementation engineer.*

Files too large to display in a browser must be opened in an editor tool.

When no records are found, the download file will contain the following XML: "<record_list ></record_list>".



After a file is removed from the server, it is not possible to recover it.

Data Table Change Log

BigMachines allows you to verify changes that have been made to the data tables through Bulk Download. This functionality will allow you download change logs from a particular data table, within a particular date and range, and according to some criteria (i.e. column x = XHDF). The change log files are downloaded as .CSV files. You may keep up to 90 days of data table history.

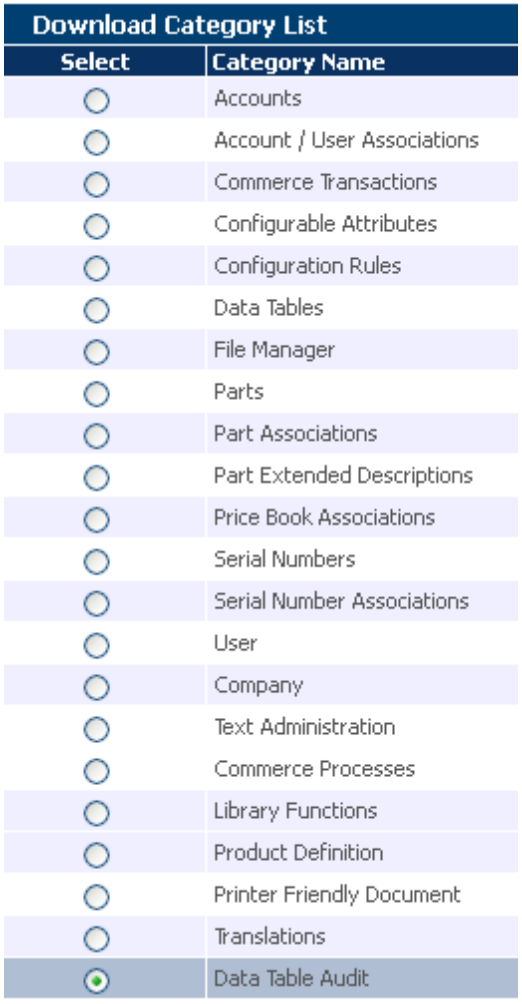


This is a paid feature. Please contact your BigMachines Account Executive for pricing and feature enablement.

Example Use Case: You work for ABC Widget Company. You have negotiated special pricing on certain components from XYZ Corporation, however these price changes are introduced periodically. You must be able to prove to XYZ Corp. that the pricing was updated at the appropriate date and time

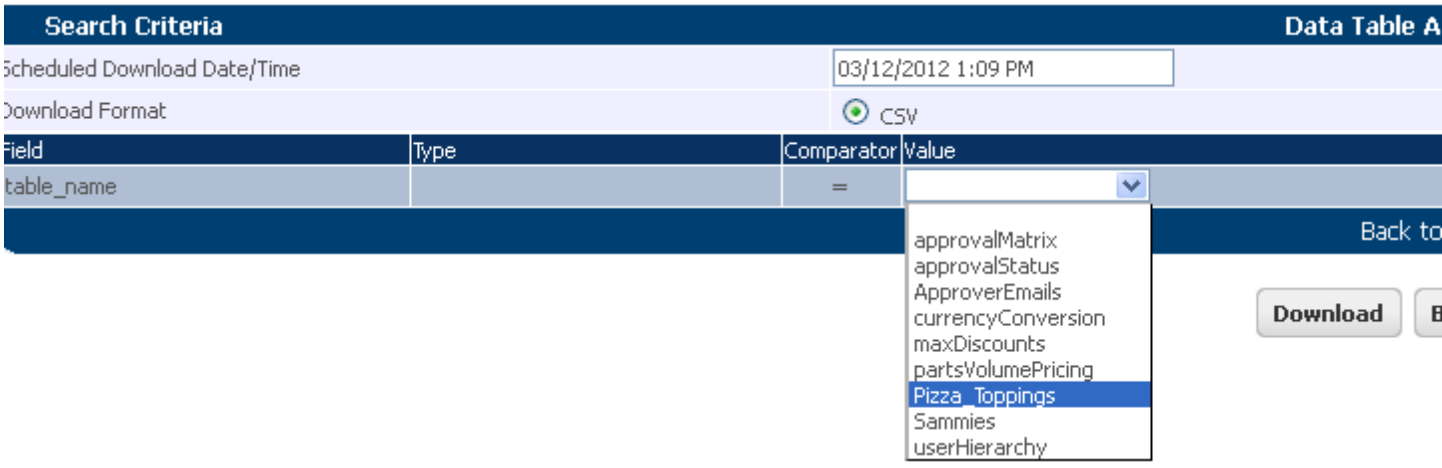
Viewing Data Table Change Logs

1. Navigate from: Admin Homepage > Utilities > Bulk Download.
2. Select **Data Table Audit** from the **Download Category List**.
3. Click **Next**.

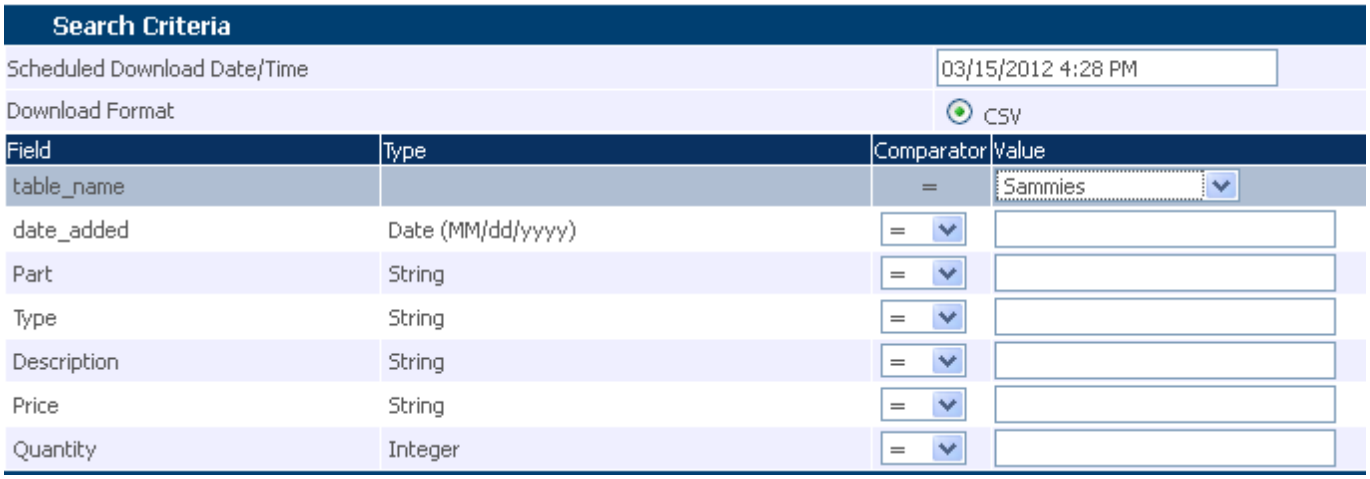


4. Schedule a date and time to download the Data Table Audit. *This file is only available for download as a CSV file.*

5. Choose the data table name from the drop-down menu. *You need this to download the Data Table Audit file.*

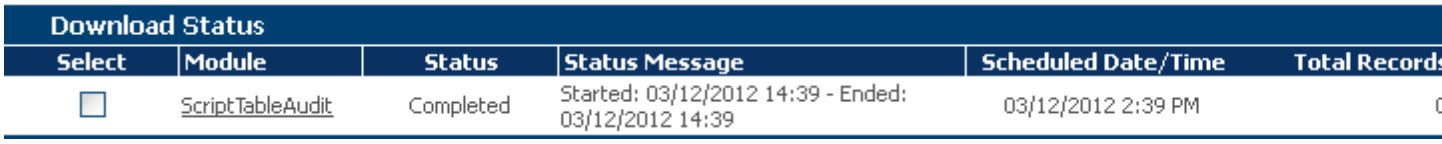


6. Select to filter change log data, if desired. *You can filter based on the table columns as well as the date the table was added.*

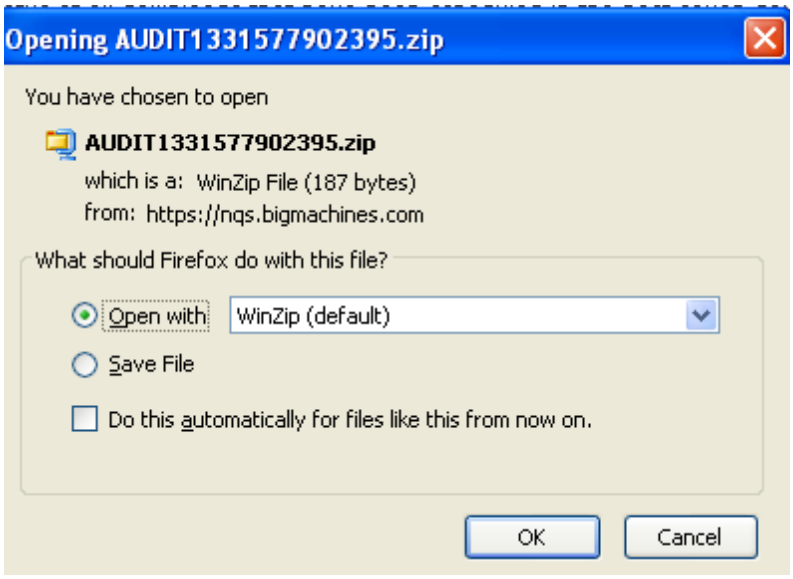


7. Click **Download**.

8. Double-click on the **ScriptTableAudit** module from the Download Status page.



9. Open the zip file.



10. Open the file and review the change log. *The file will only show you table rows that have been changed and who changed it.*

6	HS001	Hot	Meatball Sandwic	7.95	1	INSERT	superuser
7	HS002	Hot	Grilled Chicken Sa	8.95	1	INSERT	superuser
8	HS003	Hot	Italian Beef	6.95	1	INSERT	superuser
9	HS004	Hot	Grilled Veggie Sar	4.95	1	INSERT	superuser
10	CS001	Cold	Tuna Sandwich	5.5	1	INSERT	superuser
11	CS002	Cold	Bacon Turkey Clut	7.65	1	INSERT	superuser
12	CS003	Cold	Egg Sald Sandwich	8.45	1	INSERT	superuser
13	CS004	Cold	Cold Cuts Coldie	6.5	1	INSERT	superuser
14	CS004	Cold	Cold Cuts Coldie	6.5	1	DELETE	superuser
15	WS001	Warm	Bacon Turkey Brav	5.99	1	INSERT	superuser
16	WS002	Warm	Philly Steak and C	5.99	1	INSERT	superuser



Use filters to download only the information you need. *Do NOT turn this feature on unless it is necessary due to reduced data table performance.*



Log a case with BigMachines Support Portal to have the data table audit functionality enabled. *Turning on this feature may negatively affect the performance of bulk data table uploads.*

An audit record is created for every change. In this way, if you filter on a particular column value, you'll see all of the manipulations on that row. *Column data type changes are not recorded in the data audit log file. So, if you make changes to the data type of a data table column, these changes will not be reflected in the change log.*

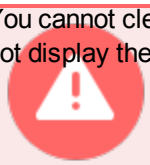
Each modification to a data table record will result in an audit record, or a row in the change log. This means each individual row that is inserted, modified, deleted or deployed will result in a row in the change log.

If you are using data table import and export (or bulk upload/download), the logs will appear as if all data is deleted and re-added. **Example:** you have a data table with 100 rows. If you delete all records and bulk upload a new table, you will see 200 rows in the change logs.



Data beyond 90 days old will not be found in the data change log.

You cannot clear a data table change log. The change log will show you what was changed, but it will not display the previous values.



RELATED TOPICS

BULK UPLOADS

OVERVIEW

Bulk Upload is used to enter large sets of data into the BigMachines system. All data that can be uploaded into the system can also be downloaded, and more importantly, all downloaded XML data can be uploaded. *You can therefore, use the download process to generate upload templates for each type of data.* The upload process enables you to add, update or delete different types of data.

Using FTP or the UI upload, you can migrate XML/CSV upload data from your local system to an internal server. The process involves generating an XML/CSV formatted file, transmitting that file (and any supporting files) to the server dedicated to receiving upload files. After the file transfer completes, you can schedule the actual upload of data through the Bulk Data Services tool. Another upload procedure is available by which the user can upload a zip file directly through the UI in the Upload section of the Bulk Data Services tool. *Depending on the number of records being uploaded, an upload process may take a few seconds to a few hours to complete.*

After the bulk upload process finishes, a bulk upload error log is generated containing a summary of the upload process, including a list of error messages. Your BigMachines fullaccessuser also receives email notice when an upload process occurs. The automatically generated email contains a summary of the upload, including number of records successfully uploaded and number of failures. Detailed errors only appear in the bulk upload error log.

Data Types Available for Upload

- Commerce Transaction
- Accounts/User Associations
- Configurable Attributes
- Configuration Rules
- Parts/Part Associations/Pricebook Associations
- Serial Numbers/Serial Number Associations
- Users
- Global Scripts
- Commerce Process
- Models
- Product Lines
- File Manager
- Translations
- Data Tables

A single XML/CSV file can contain any combination of data categories. **Example:** you can upload new pricing information for parts and new pricing rules for models. Or you can add new user accounts and CRM customer records in the same XML file. There is no limit to

the number of records you can upload at the same time. *The more records included in an upload file, the longer it will take for the process to complete, however, individual records are visible on the system as soon as they are uploaded.*

Bulk Uploads & Natural Keys

- If a user attempts to upload duplicate records (data with the same Natural Keys are pre-existing data in the table) during an XML/CSV upload, the duplicate records will be deleted and the new record will replace the pre-existing one.
- Because CSV Upload occurs row by row, if the Natural Keys match values that already exist in the table, the record will perform an update.

Bulk Upload Actions

Action	Requirements	Example
add	All tags downloaded can be used to add new data. **Example:** `<action>add</action>`	Will add a new row/record if one does not already exist. If one does exist, an error will be thrown.
modify	All tags downloaded can be used to add or modify data. **Example:** `<action>modify</action>`	Modify will create a new part or row record if it's not found. If the part/row is found, modify will update the data.
delete	All tags downloaded can be used to delete data. **Example:** `<action>delete</action>`	Will delete all the column data from a row/record.
delete_all	All tags downloaded can be used to delete all of the data. **Example:** `<action>delete all</action>`	Will delete all row and column data.
delete_by_criteria	Only available for mass delete of Parts based on custom field value.	
update	All tags downloaded can be used to update data. **Example:** `<action>update</action>`	Update will fail if the row/record is not found. If the record is found, it will modify all the column data in the row.

General Requirements:

- The upload process supports uploading a single document (with sub-documents) only.
- The system validates all commerce and asset management data generated through the XML upload (or real-time integration) process. *Each field must contain the data type the system expects.*

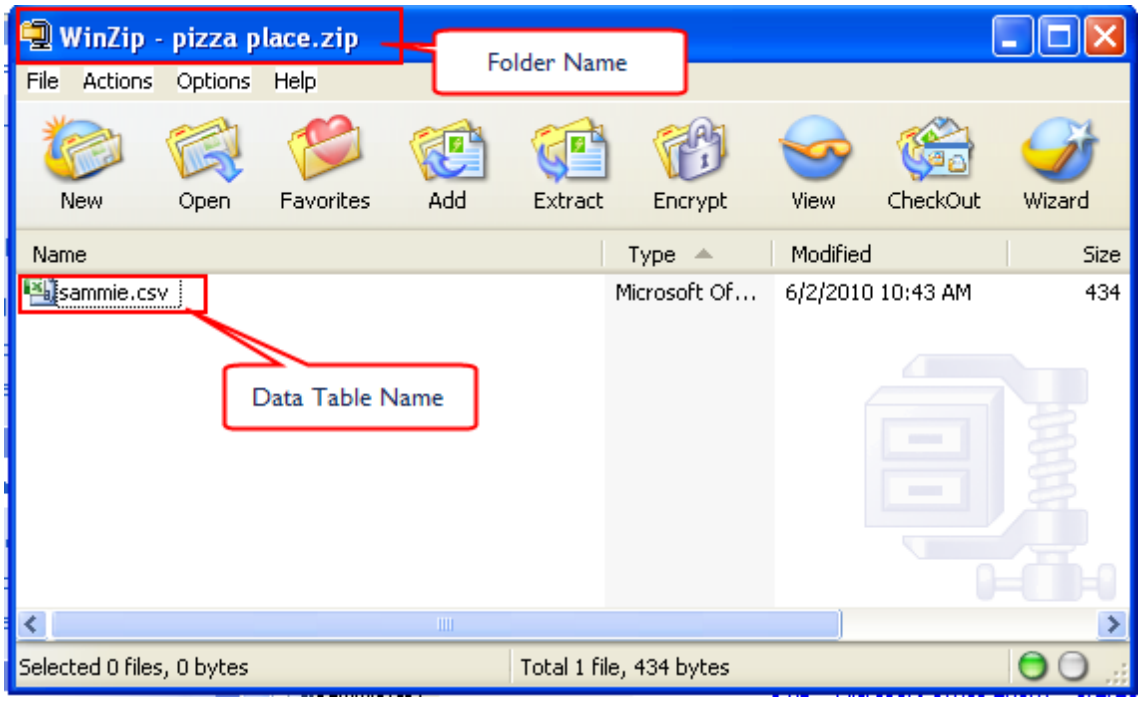


To view the full XML format for a particular process, you can use the download administration area to extract a single record from the system. When downloading commerce or asset data, the entire transaction, including all documents and sub-documents, is downloaded from the system.

Bulk Upload of CSV Data Tables

The system requires some way to determine which bulk upload the .zip files should go to. This will require you to use the tag `<type>`:

- `<type>DATA_TABLES_CSV</type>` is the type for ONLY for CSV data tables.
- `<type>REGULAR</type>` is used for ALL other bulk uploads.



For the DATA_TABLES_CSV type, the .zip name should be the BigMachines folder name that the data table is located in. The name of the .csv file within the .zip file is the actual table name.

Actions Supported by XML

Upload Category	Category Name	Add	Update	Modify	Delete	Delete All
Commerce Transaction	bm_cm_bs_data		X			
Accounts	bm_crm	X	X	X	X	X
Account/User Associations	bm_crm_user_assoc	X	X	X	X	X
Configurable Attributes	bm_config_att	X	X	X	X	
Configuration Rules	bm_config_rules	X	X	X	X	
Parts	bm_part	X	X	X	X	X
Parts Associations	bm_parts_associations	X			X	X
Part Extended Description	bm_part_extended			X		

Upload Category	Category Name	Add	Update	Modify	Delete	Delete All
Serial Numbers	bm_serial_number	X	X	X	X	X
Serial Number Associations	bm_part_in_sn	X			X	X
User Accounts	bm_user	X	X	X		
Data Tables	bm_script_data	X	X	X	X	X
Commerce Processes	bm_cm_version	X	X			
Product Lines	product_line	X	X	X	X	X
Models	model	X	X	X	X	X
Price Book Associations	price_book_variable_name	X	X		X	
File Manager	bm_shared_file	X	X	X	X	X

Required XML Tags for Upload



XML Required Tags change every release. *The easiest way to obtain an XML template that is current is to download an item of the same category.*

Tag	Tag Function
<record_list>	Starts and ends the file.
<each_record>	Defines the boundaries of a single record (for example, user record, part record, etc.)
<category>	Determines the type of data included in the record (for example, bm_customer, model, etc.)
<action>	Informs the system what to do with the record, either add, update, delete or delete all.

Required Structure for Single XML Record

```
<record_list>
<each_record>
<category>[category name]</category>
<action>[action name]</action>
<... category specific tags ...>
<![CDATA[ etc. etc. etc. ]]>
</ ...>
</each_record>
</record_list>
```

Required XML for a Line Action

```
< each_record >
< category > bm_cm_bs_data </ category >
< action > update </ action >
< id > <![CDATA[ 9831574 ]]></ id >
< data_xml >
< document document_var_name =" quote " data_type =" 0 " >
< sub_document >
< document document_number =" 3 ">
< _price_quantity ><![CDATA[ 10 ]]></ _price_quantity >
< applicationcomments_quoteline > test -Another2 </ applicationcomments_quoteline >
</ document >
</ sub_document >
</ document >
</ data_xml >
< action_data >
< action_var_name > update_quoteline </ action_var_name >
</ action_data >
</ each_record >
```

Required XML for Quote Action

```
< each_record >
< category > bm_cm_bs_data </ category >
< action > update </ action >
< id > 20134654 </ id >
< data_xml >
< document document_var_name =" order1 " >
</ document >
</ data_xml >
< action_data >
< action_var_name > accept_order </ action_var_name >
</ action_data >
</ each_record >
```



The site version must be included in the tag `<site_version>`. *You can download XML from a site of one version and upload it to a site of the same version.*

Image files must have .pdf, .gif, .png or .jpg extensions. Most file types are supported for non-image file



attachments



All values are trimmed by the system, including leading and trailing spaces.

The default value for all fields with a specified value is an empty string.



For each record, if a warning or error occurs, the warning or error will be displayed along with the XML data record that threw the error.

ADMINISTRATION

Uploading Data Using UI

1. Create a single XML/CSV file containing the records you want to create, update, or delete.
2. Organize all referenced attachment files into a folder called "img", and put that folder into the same directory as the XML file.
3. Zip up the XML file and all attachments into a single zip file. The zip file must not contain any path names.
4. Log in to the BigMachines system.
5. Navigate to the **Bulk Data Services** section under the Admin tab.
6. Click **Upload**.
7. *The file just uploaded should appear on the Upload Files List page.* Do either of the following:
   - **To start upload process immediately:** Mark the checkbox corresponding to the filename, and click Upload.
   - **To schedule upload at a future time:** Mark the checkbox corresponding to the filename, enter an upload time, and then click Upload.
8. Review the uploaded task on the **Upload Status** page. *The task will display a "pending" status until the upload begins. Finished uploads display a "completed" status.*



After the upload process occurs, view the error log.



After the bulk upload process finishes, a bulk upload error log is generated containing a summary of the upload process, including a list of error messages. Your BigMachines fullaccessuser also receives email notice when an upload process completes.

The automatically generated email contains a summary of the upload, including number of records successfully uploaded and number of failures. Detailed errors only appear in the bulk upload error log.

Uploading Data Using FTP

1. Open a ticket for OPS to create an FTP profile for you and the "automated" folder on the FTP server. *Inform OPS of the application URL and company/client name.*
2. Download the XML files from the application that you will be updating (parts, parts extended descriptions, data tables, etc.).
3. Fill the XML with data. Each XML file that needs to be run needs to be placed in a separate .zip file by itself. *The upload_list.xml file should fill in the* `<file>` *nodes with the name of the .zip files to be uploaded.* **Sample Format:**

```
<? xml version="1.0"?>
<record_list>
<each_record>
<category>bm_script_data</category>
<table_name>CustomerAddress</table_name>
<action>deploy</action>
</each_record>
</record_list>
```

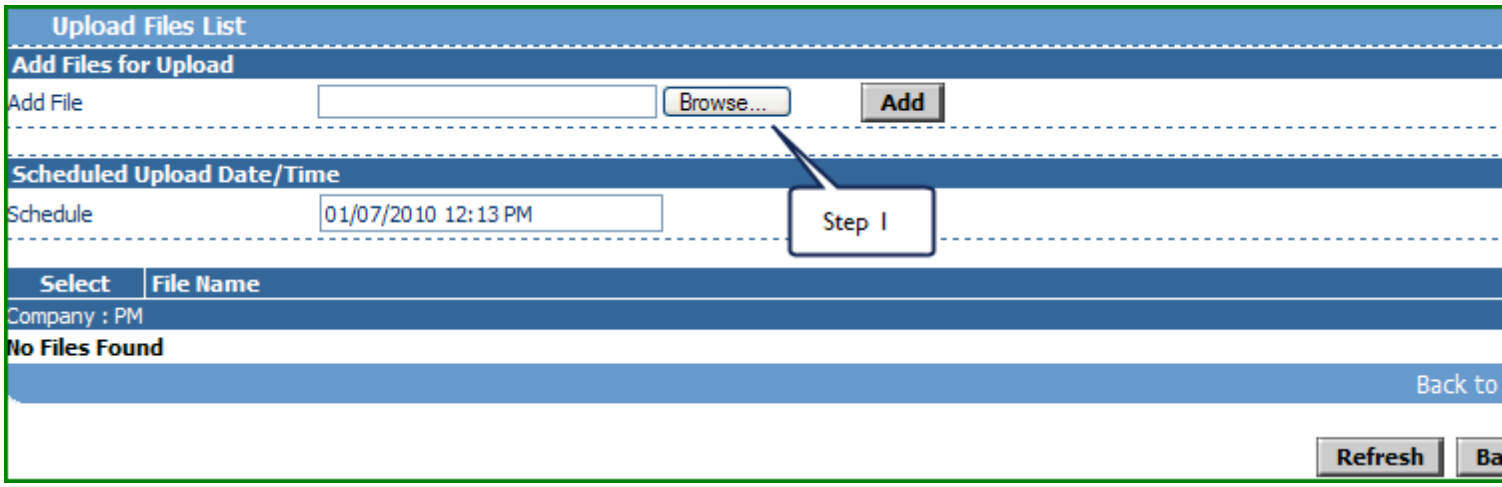
4. Manually test the XML files.
    - **Example:** Navigate from: Admin > Utilities > Bulk Data Services > Upload- *attempt to load a zipped XML file to ensure it works.*
5. Submit the files to the FTP location supplied by OPS. *Ensure the directory is in the "automated" folder (this folder is run on a timer to pull files) and submit the ZIP files first.*
6. Upload the `upload_list.xml` file last. *This file will execute and run the .zip files in the order they are listed.*
7. Navigate from: Upload Files list page to confirm that the Upload has taken place.

Using FTP, you can migrate XML upload data from your local system to the BigMachines server. *The process involves generating an XML-formatted file, transmitting that file (and supporting files) to the BigMachines server dedicated to receiving upload files.*

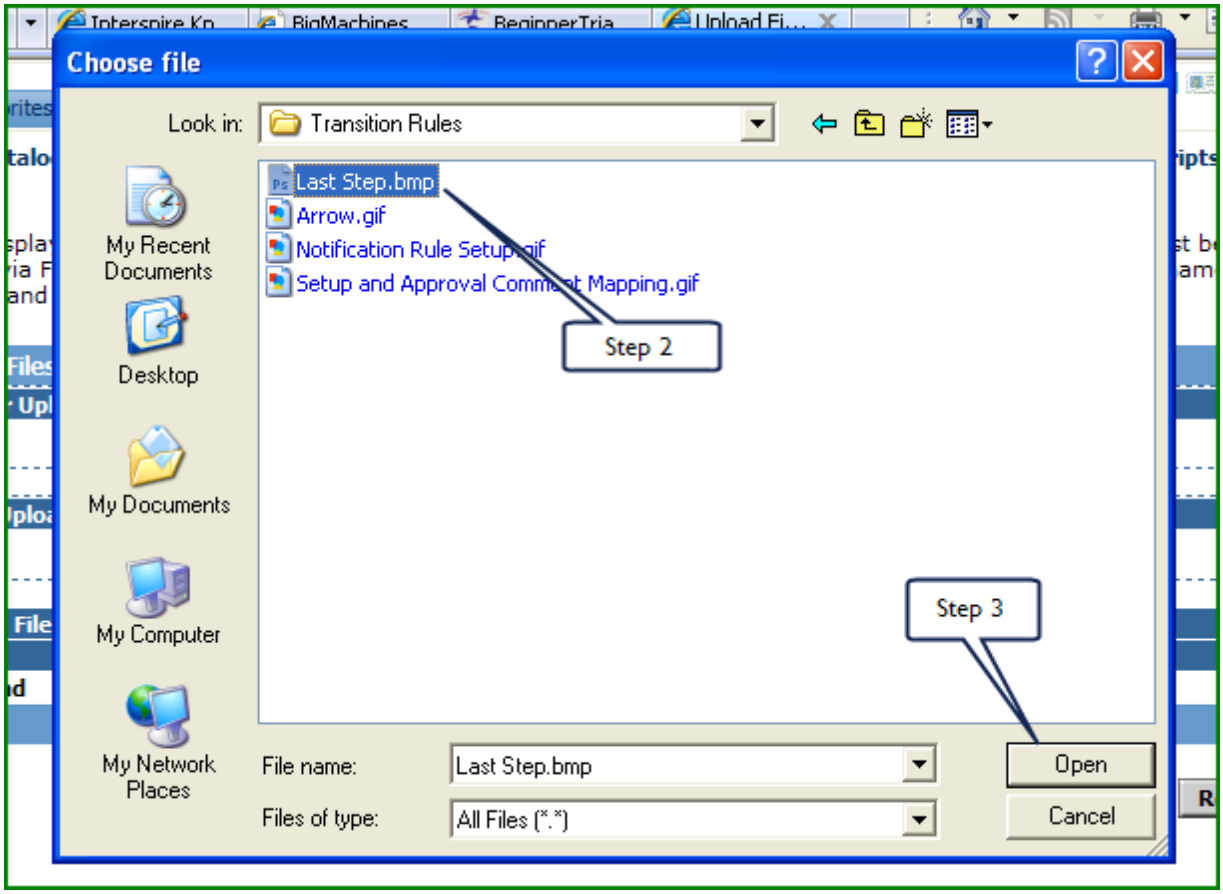
After the file transfer completes, you can schedule the actual upload of data through the Bulk Data Service tool. *Depending on the number of records being uploaded, an upload process may take a few seconds to a few hours to complete.*

Adding Files for Upload

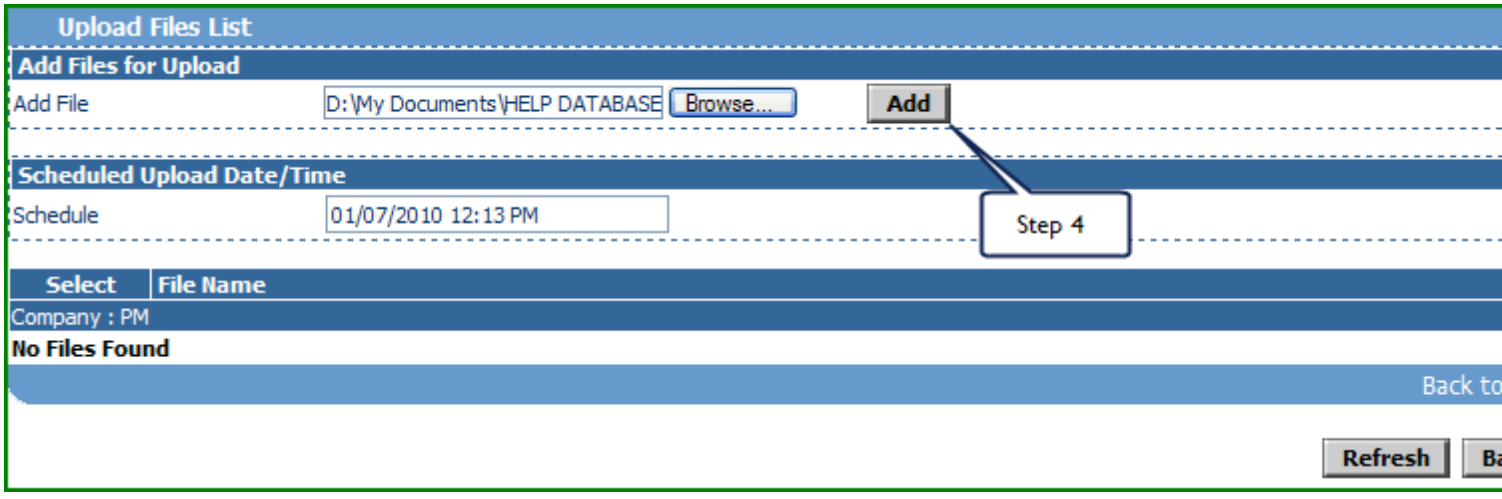
1. Click **Browse** next to the **Add File** field. *This opens the File Upload window.*



2. Select the file you wish to upload.
3. Click **Open**. *This displays the name of the selected file in the Add File field.*



4. Click **Add**.



Scheduling an Upload

1. Enter the date and time in `mm/dd/yyyy hh:mm` format against the Schedule field.
2. Click **Refresh** to add the new file.

Viewing Status of Upload

Enables you to view the status of all upload tasks scheduled during the previous week. Successful uploads showing a "completed" status are periodically removed from the upload status page every 7 days. Upload tasks with any status other than completed are not removed from the status page. The Upload Status page shows the **file name** and **status** of all files scheduled for upload in the previous 7 days. The page also displays the **status message**, the **scheduled date and time**, and options to open the **Log File**, refresh the information on the page, stop the currently running tasks, and delete existing information. **Available actions from the Upload Status page:**

- Sort the details based on **Status**, **Status Message**, **Scheduled Date/Time**, and **Log**. *Click the appropriate heading to sort the details in the order you wish to view.*
- View error logs for a file by clicking the log link against it. *This displays a new window with all the log entries for the errors and warnings resulting from the upload process selected.*
- Stop an upload process by clicking **Stop Running Task**. Uploads with a status of "running" can be stopped. *To stop a currently running upload mark the checkbox corresponding to the upload filename you want to stop, and click **Stop Running Task**.*
- Remove listings for old uploads by selecting the files to be deleted and click **Delete**. Uploads with a status of pending or completed can be deleted from the **Upload Status** page. *To prevent an upload from happening in the future, mark the checkbox corresponding to the upload filename you want to cancel, and click Delete.*

Status	Description
Pending	Pending uploads are awaiting scheduling. As soon as a file is scheduled for upload, the uploaded (via FTP) Zip file is automatically deleted from the system.
Loaded	Loaded uploads means the file is in the queue. Only one upload process can run at a time, so when multiple uploads are scheduled, the system loads them into a queue and then processes each upload file one at a time.
Running	Running indicates the file being uploaded. This status can only display for a single upload file at a time because only one upload process can happen at a time.
Completed	Uploads showing a completed status have finished. You can view the error logs corresponding to completed uploads to verify the success each upload process.
Failed	This status indicates that either the user stopped the upload process or a system error occurred. *When a user stops an upload that is currently running, the data uploaded up until that point is not backed out of the system.*



If an upload fails, retry the process . *If you cannot get a upload to succeed, contact an implementation engineer.*

If you want to wait for an upload process to complete, you can periodically refresh the Upload Status



page by clicking Refresh.



Upload tasks can have any of the following status: pending, loaded, running, completed or failed. Pending means the upload is awaiting scheduling. *As soon as the file is scheduled for upload, the Zip file is deleted from the system.*

Data already uploaded into the system is not backed up when an upload process is stopped.

Deleting Parts by Criteria

You can delete multiple parts by XML upload based on Parts Field Values.The user can use the `delete_by_criteria` action in the upload XML to specify the criteria for which the parts must be deleted. The criteria is defined by specifying the part field, the value of the field and the comparator with which the comparison should be made. When the criteria is satisfied, the part is deleted.

Parts Fields available for use

- Part Number
- Direct Buy
- Description
- Units
- Lead Time
- All deployed part custom fields *(except multi-select menu types)*

Allowed Comparators

- =
- LIKE
- <
- >
- >=
- <=
- <>

Sample XML: delete_by_criteria Action

```
<!-- Specify Action Name -->
<action>delete_by_criteria</action>
<!-- Specify Criteria -->
<criteria>
<!-- Specify Field Name, Value and comparator. In this example, delete all parts where _part_custom_field2 =
test -->
<field>_part_custom_field2</field>
<value>test</value>
<comparator>=</comparator>
</criteria>
```

Sample XML: Like Comparator

```
<!-- Specify Action Name -->
<action>delete_by_criteria</action>
<!-- Specify Criteria -->
<criteria>
<!-- Specify Field Name, Value and comparator. In this example, delete all parts where part number contains
"abc" -->
<field>part_number</field>
<value>%abc%</value>
<comparator>LIKE</comparator>
</criteria>
```



The Like comparator can only be used for string type field . *To use this comparator, the user has to specify the wildcard "%" in the value.* **Example:** the above script will delete all parts that have abc in their Part Number.

Deleting Parts with Multiple Criteria

Multiple criteria can be declared within `each_record` tag and will have an implicit **AND** between them. If the user wants to achieve criterion's **OR**, use multiple `each_record` tags and specify the **OR** criteria in different `each_record` tags. **Example:** you want to delete all Parts where `_part_custom_field2 = "Domestic"` and lead_time is greater than 5. This can be accomplished by uploading an XML where the `delete_by_criteria` action allows the user to specify the two criteria for which the parts must be deleted.

Sample XML: Deleting Parts by Multiple Criteria

```
<!-- Upload XML to delete Parts when the lead time is greater than 5 AND _part_custom_field2 = "Domestic"-->
```

```
<?xml version="1.0" encoding="UTF-8"?>
<record_list>
<!-- Provide appropriate site version -->
<site_version><![CDATA[9.2]]></site_version>
<each_record>
<category>bm_part</category>
<action>delete_by_criteria</action>
<criteria>
<field>lead_time</field>
<value>5</value>
<!-- All comparators with the exception of "=" and "LIKE" must be wrapped in CDATA tags -->
<comparator><![CDATA[ >]]></comparator>
</criteria>
<criteria>
<field>_part_custom_field2</field>
<value>Domestic</value>
<comparator>=</comparator>
</criteria>
</each_record>
</record_list>
```

TROUBLESHOOTING

Troubleshooting

Cases	Add	Update
column missing	default value	leave value unchanged
blank cell (in Excel)	default value	leave value unchanged
$EMPTY$	treat as blank value	treat as blank
required fields validation	throw error	ignore if value available in database
IGNORE action	ignores an entire row of data	ignores an entire row of data



Error messages in parts/pricebook export sometimes refer to a record number. If you are viewing the CSV of the list being uploaded, knock off the top three rows from this number, and the number will point straight to the offending record. You can try to just delete the top three rows in excel, so that the row count matches up to the error in BigMachines. To upload a blank field in



CSV: $EMPTY$

Reading Error Logs

The Upload error log reports errors that occur during the bulk upload process. The upload error log header contains the number of items uploaded successfully, the number of items that experienced some type of failure, and the time it took for the upload process to complete. Specific errors appear below the log header. *Error messages are severe in that they prevent specific records from being uploaded. When errors occur, the log file reports the failed record's data and a reason for the error.*

Warning messages display below the list of error messages. The difference between a warning message and an error message is its severity. Warnings call out minor errors that do not prevent an item from being uploaded. **Example:** if a part references an image file that is not included in the zip file, the part will be uploaded, but the image will not be associated with it. Similar to error messages, warning messages contain information to help you pinpoint the problem, such as the line number of the record containing the warning, the cause of the warning and a suggested solution.

How do I correct a problem reported in an upload error log?

You can correct error log warnings and errors in several ways. If the errors are due to a problem in the XML file, then the XML file can be corrected, re-zipped, and re-uploaded via FTP. When the error is due to an image file, then the image file can be re-zipped (with an XML file), and re-uploaded. Another option is to use the BigMachines system's administration area to manually correct warnings and errors.

The method you choose to correct error log problems should take into consideration the number of problems that need to be fixed. If many problems need to be fixed, then manually entering the information could be a time consuming process. In such cases, it is often quicker to fix the errors directly in the XML files and then re-upload everything.

Why are there so many errors logged?

The error log is there to help in debugging problems. Every time an "illegal" input is applied to a field, the error is caught by a lower level of the BM engine and propagated up to the surface before it is displayed to the user. **Example:** if a user were to enter the string "Two" where a number was expected, an error would be caught on the server side and an error was would be passed back along with the stack trace, the nested calls to lines of code within the server-side of BM.

This is all very complex and meant to be kept transparent. The end result of this is an error message like "Encountered " ")" ") "" at line 1, column 81. **Was expecting one of:**

```
<NUMBER> ...
<STRING> ..."
```

to be displayed towards the top of the window. This is mirrored, along with the stack trace, in the error log. If a user were to use trial and error, and enter "Two", then "two", then "TWO", then "2", 3 errors with stack traces would be produced adding up to possibly 500 lines in the error log.



Check this file after completing a bulk upload process to verify its success.

NOTES



The delimiter for CSV data is a comma.



Upload of Multi-byte characters is not supported for CSV.

RELATED TOPICS

FAVORITES LIST

OVERVIEW

Favorites allows each user to list their favorite product and/or parts for quick access and quote creation in a Favorites List. Favorite lists are maintained independently of catalog and parts. Because of this separation, products in the Favorites list can become out of sync with the catalog products/parts they reference. The update functionality to ensure that a user's Favorites list is synchronized with the catalog and parts. These updates ensure that products on the Favorites List display current catalog information. The update engine verifies product attributes, pricing, and availability.

The update engine can only calculate pricing information on products with still-supported configurations. When products have new or modified configurable attributes, you can set the engine to skip over them during the update process. This speeds up Favorite List updates, but it also forces users to reconfigure those products before purchase.Favorites List updates display to the user in several ways. *Certain updates appear without any user notification, like pricing and product attribute changes. Other updates change the status of items on the Favorites List, such as products that are no longer available or that must be reconfigured.*

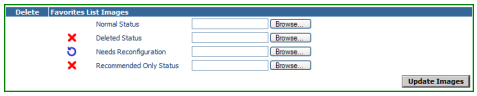
Status Values

A status column on the Favorites List informs users whether products have changed since they were added to the list. Status is represented graphically with or without (for normal status) an icon. You can customize the status icons that display.

- **Normal** - Items that can be directly purchased from the Favorites List have a Normal status. There is no system-defined icon for this status. If you want to display an icon alongside products with a Normal status, you can associate an icon with the status. *The Normal status displays even if a product's price or regular attributes have changed because those types of changes do no affect whether the product can be purchased.*

- **Delete** - Items that are no longer available for purchase because they have been removed from the catalog or changed from a direct buy to a non-direct buy have a Delete status. The system notifies the user of this status with a delete icon.

- **Reconfiguration** - Items that require user reconfiguration before purchase have a Reconfigure status. The system notifies the user of this status with a reconfigure icon.



You can verify Favorites List updates by viewing the email notification generated when an update process finishes. The system sends a confirmation email to the email address specified for the update. Confirmation emails contain the number of products processed, the number of items that no longer exist in the catalog, and the number of items that now need to be reconfigured before purchase.

Sample Email Contents

```
Number of items processed : 118

Number of items flagged for delete : 2

Number of models flagged for reconfiguration : 10

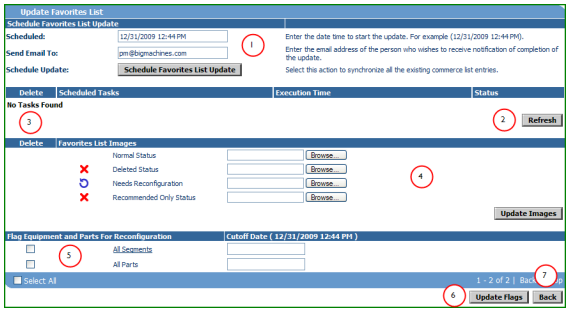
Number of parts flagged for reconfiguration : 0
```

ADMINISTRATION

Managing Favorites List

This administration page is used to update the product data displayed on the Commerce List. You can tag items for reconfiguration or change the status icons the system uses to indicate changes to the Commerce List before performing an update.

1. Schedule a Favorites List Update:
    - Enter the time for starting the update in the **Scheduled** field. **Example:** (11/11/2009 11:11AM).
    - Enter the email address of the person who wishes to received notification of completion of the update in the **Send Email To** field.
    - Click **Schedule Favorites List Update** to synchronize all the existing commerce list entries.
2. Sort the **Scheduled Tasks** list alphabetically by column.
    - To sort products: choose the column you want to sort by and click the header.
    - To reverse-sort: click the header twice.
3. Select the **Scheduled Task** to delete by clicking the checkbox in the **Delete** column and clicking **Refresh**. *You can delete the scheduled task to cancel the process before a Favorites List update begins. While an update is happening, it is not possible to stop the process.*



4. Click **Browse** to select the Favorites List Images. Upload the image.
5. Select the items that should not be updated in the **Flag Equipment and Parts for Reconfiguration** section. *To do so: navigate to the items that have different supported configurations by clicking a Link and mark all parts for reconfiguration; or navigate through the products hierarchy and mark select products.*
6. Click **Update Flags** on each page with selected items to save your selection.
7. Click **Back** to return to the admin home page.

NOTES

You can schedule Favorites List updates to run whenever you like; ideally, they should be run during your system's low usage times.



The update functionality represents status graphically on the Favorites List.

However, it is the user's prerogative to delete items that are no longer available/ reconfigure items that have been marked for reconfiguration from the Favorites List.

RELATED TOPICS

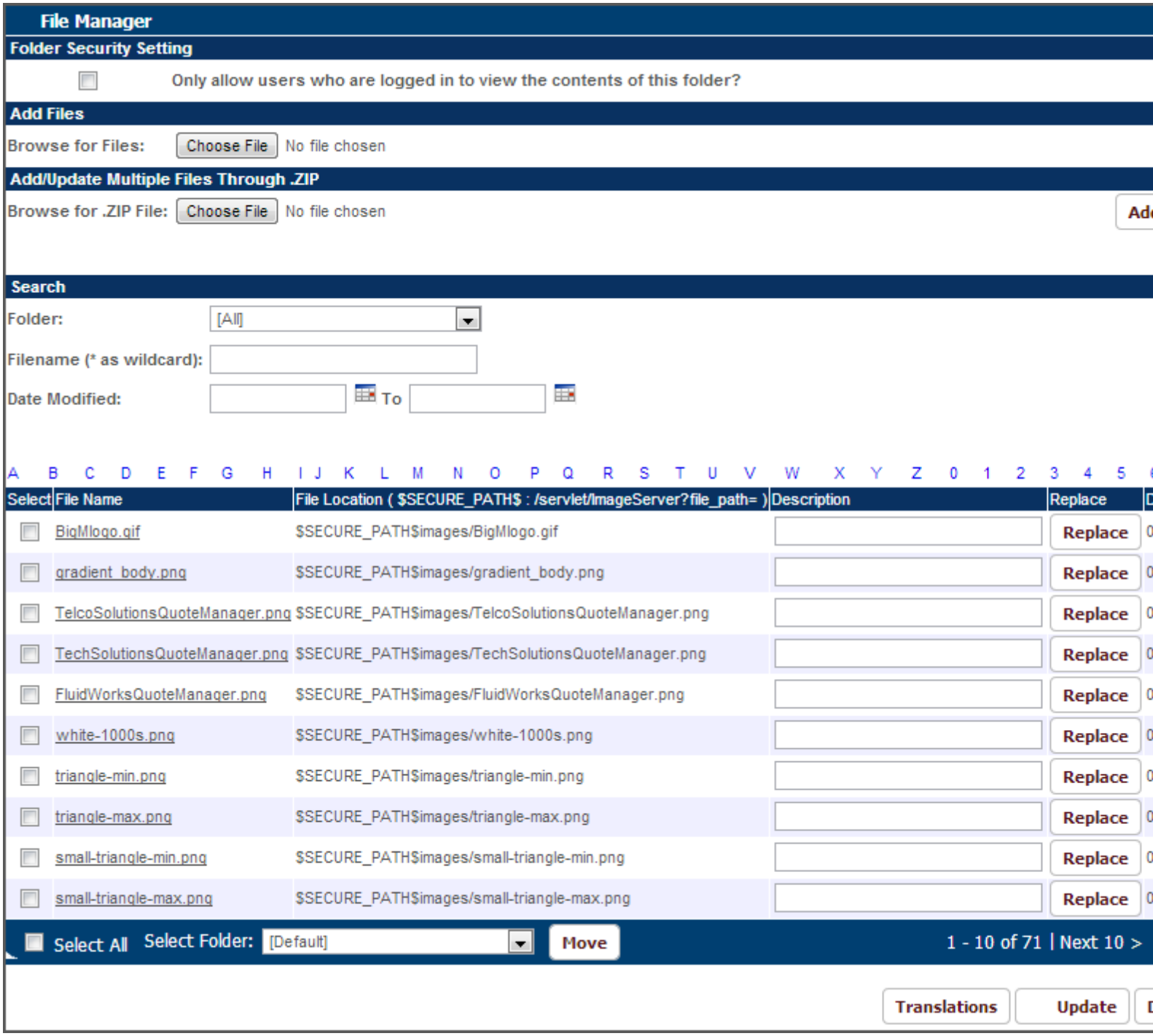
FILE MANAGER

OVERVIEW

An integral part of BigMachines is the File Manager. You can upload any kind of file to the File Manager, organize them into folders and access them from anywhere on the Internet. **The File Manager is used for:**

- External images
- JavaScript files linked to various areas on the site
- CSS Stylesheets for Configurations Flow Templates
- Excel spreadsheets (to hold master data)

Example of File Manager



Pagination functionality and categorization of files by alpha-numeric indices allows you to easily navigate through a large numbers of files. Search and paging functionality are available within image selector pages in configuration and commerce. **File Manager allows you to:**

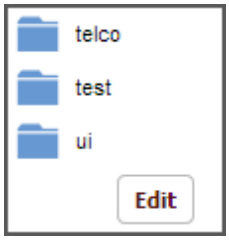
- Search by File Name (including support for * as a wildcard)
- Search by Last Modified Date Range
- Search for files within a folder or across all folders

ADMINISTRATION

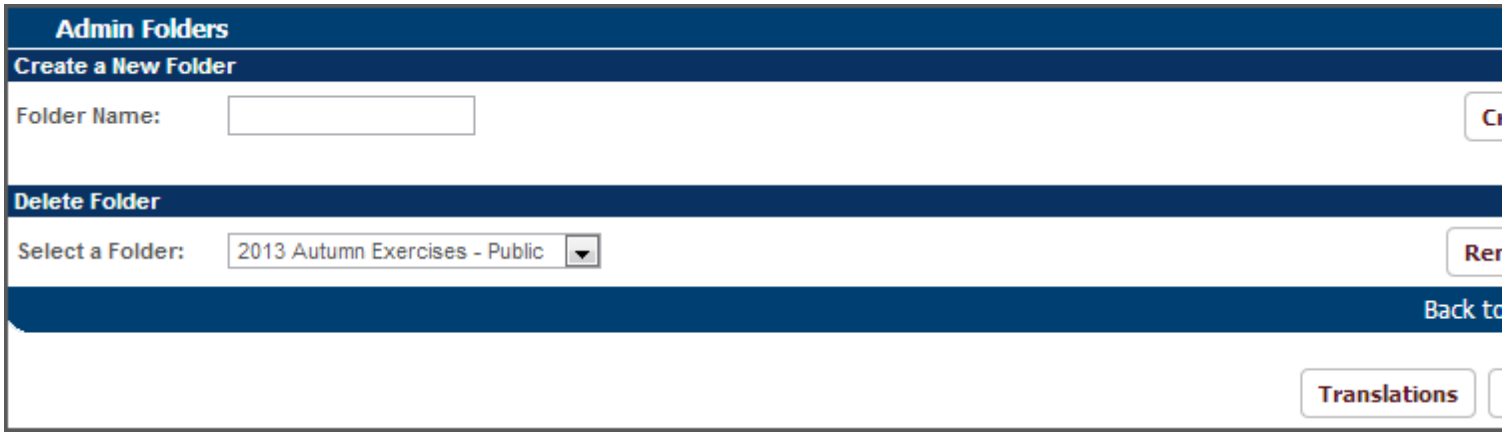
Bulk Upload Actions:

Adding a Folder

1. Navigate from: Admin Home Page > Utilities > File Manager
2. Click **Edit** at the bottom of the Folders list.



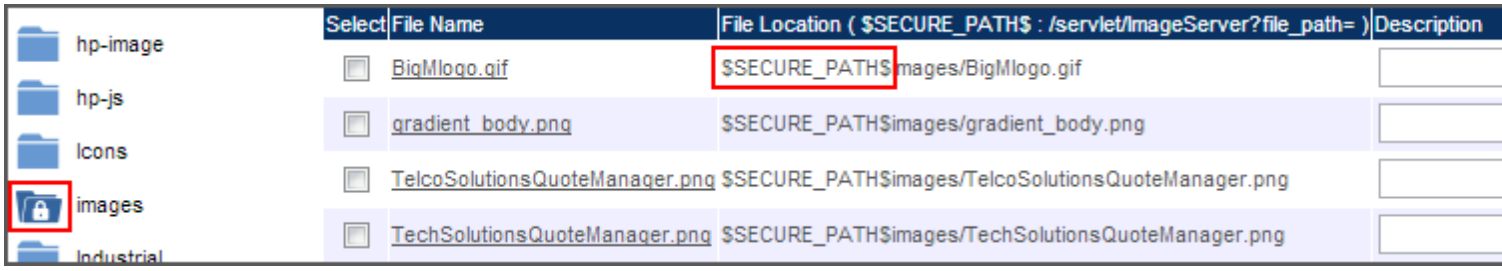
3. Enter a **Folder Name**.



4. Click **Create**.
5. **Optional:** If you wish to make a folder secure, select the **Folder Security Setting** checkbox and click **Save**.



6. Once a folder is designated as secure, the File Manager files and folders will appear as such:



7. Once finished, click one of the following:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Update:** Applies changes to File Manager.
   - **Delete:** Deletes new content and changes.
   - **Back:** Returns user to previous page.

Only folders can be designated as secure; the administrator can designate any folder they have added as secure. *Once an administrator has designated a folder as secure, the security settings will apply to all files within that folder.*

If you attempt to add a file, and the folder name does not exist, then the system creates the folder and the file in it. *The File Manager Bulk Upload does not support the creation of a folder alone, without any files in it.*

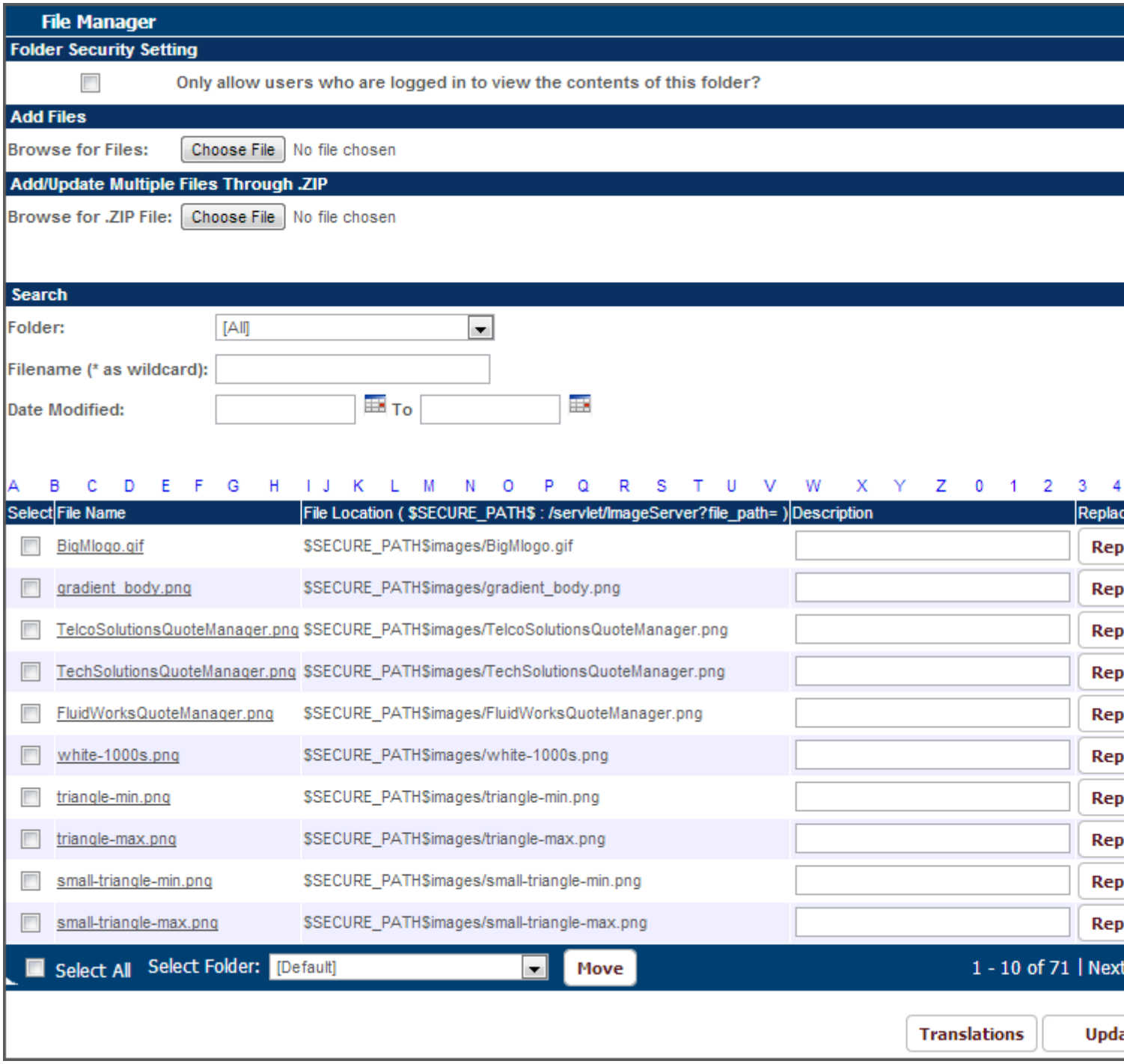
If you add a file where the folder exists, the system adds the file in the existing location.

If you add a file and the file name already exists in the folder, then the system throws an error.

User cannot perform a bulk change of a folder's security settings.

Adding a File

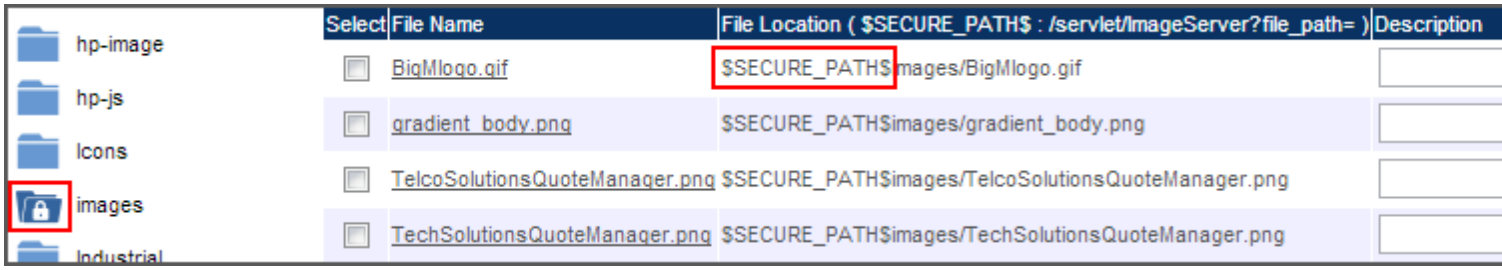
1. Navigate from: Admin Home Page > Utilities > File Manager.
2. Locate the **Folder** that you would like to add a file to.
3. Once in the folder, choose whether you would like to **Add Files** or **Add/Update Multiple Files Through .ZIP**.



4. Click **Choose File** to select the file(s) you would like to add to the folder.
5. Click **Add File** or **Add/Update Files** to add the selected files to the folder.
6. **Optional:** If you wish to make a folder secure, select the **Folder Security Setting** checkbox and click **Save**.



7. Once a folder is designated as secure, the File Manager files and folders will appear as such:



8. Once finished, click one of the following:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Update:** Applies changes to File Manager.
   - **Delete:** Deletes new content and changes.
   - **Back:** Returns user to previous page.



Only folders can be designated as secure; the administrator can designate any folder they have added as secure. *Once an administrator has designated a folder as secure, the security settings will apply to all files within that folder.*



If you attempt to add a file, and the folder name does not exist, then the system creates the folder and the file in it. *The File Manager Bulk Upload does not support the creation of a folder alone, without any files in it.*

If you add a file where the folder exists, the system adds the file in the existing location.

If you add a file and the file name already exists in the folder, then the system throws an error.

User cannot perform a bulk change of a folder's security settings.

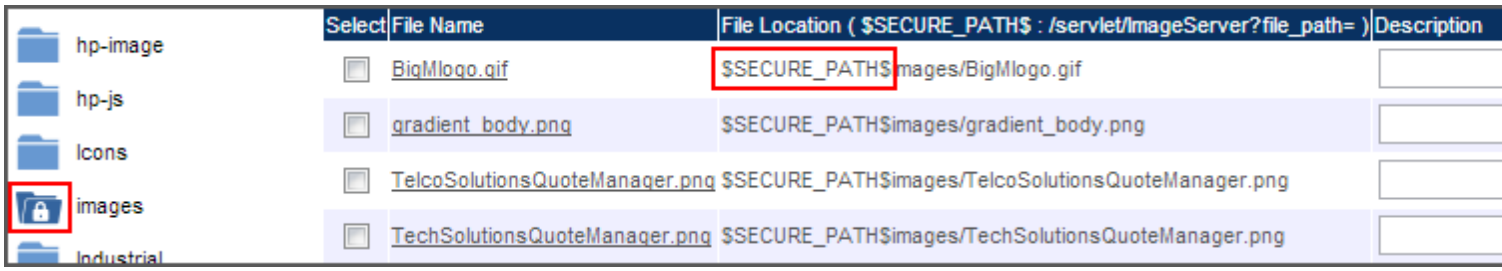
Available Actions:

Update

1. Navigate from: Admin Home Page > Utilities > File Manager.
2. Locate the folder or file you would like to update.
3. Click **Update**.
4. **Optional:** If you wish to make a folder secure, select the **Folder Security Setting** checkbox and click **Save**.



5. Once a folder is designated as secure, the File Manager files and folders will appear as such:



Only folders can be designated as secure; the administrator can designate any folder they have added as secure. *Once an administrator has designated a folder as secure, the security settings will apply to all files within that folder.*

User cannot perform a bulk change of a folder's security settings.

If you attempt to update a file, where the folder does not exist, the system throws an error.

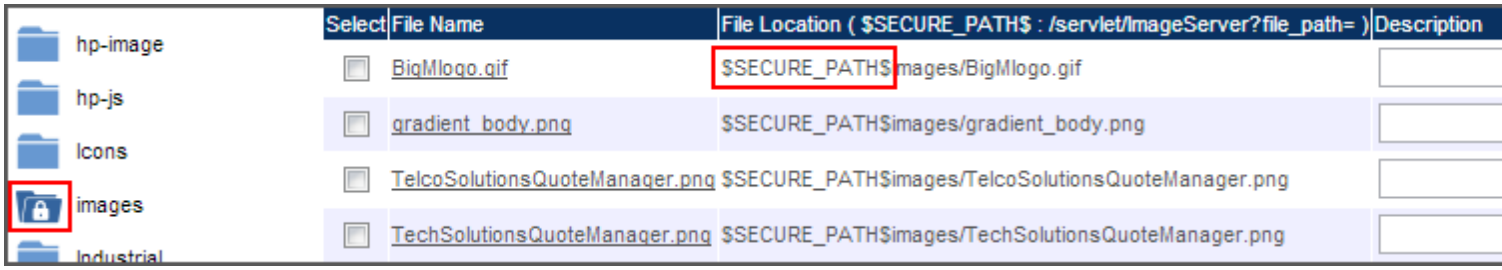
If you attempt to update a file, where the file name does not exist in the specified folder, the system throws an error.

Modify

1. Navigate from: Admin Home Page > Utilities > File Manager.
2. Locate the folder or file you would like to modify.
3. Once finished, click one of the following:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Update:** Applies changes to File Manager.
   - **Delete:** Deletes new content and changes.
   - **Back:** Returns user to previous page.
4. **Optional:** If you wish to make a folder secure, select the **Folder Security Setting** checkbox and click **Save**.



5. Once a folder is designated as secure, the File Manager files and folders will appear as such:





Only folders can be designated as secure; the administrator can designate any folder they have added as secure. *Once an administrator has designated a folder as secure, the security settings will apply to all files within that folder.*



User cannot perform a bulk change of a folder's security settings.

If you attempt to modify a file, where the file does not exist (but the folder does), then the system creates a new file in the existing location. *If the file already exists, then it will modify the file.*

If you attempt to modify a file, and the folder does not exist, then the system creates the folder and places the file in the folder. *The File Manager Bulk Upload does not support the update of a folder alone, without any files in it.*

Delete

- If you use the delete action and provide only the folder name, then the system checks for the folder name and deletes the folder (with all files in it).
- If you use the delete action for the [Default] folder, then all the files inside this folder are deleted but the [Default] folder is not deleted.
- If you use the delete action with the specific folder name and file name, then only that file is deleted from the specific location.
- If you want to delete multiple files in a folder (and not the folder), you need to use multiple delete actions.



To use the delete action for a File Manager Bulk Upload process, you must have either the file name or the folder name. *Delete action does not work in case you only use tags like description.*

Delete All

- Use this action to delete all the folders and files in the File Manager.
- This action will delete all the files in the [Default] folder, but will not delete the [Default] folder.



To use the delete action for a File Manager Bulk Upload process, you must have either the file name or the folder name. *Delete action does not work in case you only use tags like description.*

NOTES



To **Add** a folder, click **Edit** and complete Folder Name and click **Create**. To **Delete** a folder, click **Edit** and select a folder from the menu, click **Remove**.

Browse for single files or .ZIP files and click Add (or Add/Update) when you are done. *Files unzipped in this manner will replace any existing files in this folder that have the same filename.*

File Location: Copy these values to access the files from anywhere on the web. **Example URL path:** `mysite.bigmachies.com/bmfsweb/mysite/image/Branding/logo_small.gif`

When referencing a File Manager file anywhere on the site (i.e. - Header/Footer/Config HTML Templates, etc.) use the $BASE_PATH$ shorthand URL: `$BASE_PATH$/Branding/logo_small.gif`



You can replace or update existing files by clicking **Replace** next to the selected file. *Be careful when adding or removing files. The File Location changes when this happens. If the file is referenced anywhere on the site, the link will be broken.*

When performing a Bulk Download of the File Manager, both secured and unsecured files will be downloaded.

When performing a Bulk Upload of the File Manager, the user cannot specify files to be uploaded to a secure folder. *If uploading to an existing folder, the pre-existing security settings will remain. If uploading to a new folder, the folder will be unsecured by default.*

When migrating, the security settings from the source folder will be transferred to the target folder.

When updating a folder's security settings, the "Date Modified" of all the files located within the folder will be updated to the current date.



File names are case and space sensitive.

The user cannot perform a bulk change of a folder's security settings.

If the security settings on the target and source site do not match a particular file or folder, then the checkboxes will be disabled for those files. *The text "Security does not match the source" will be displayed next to the folder whose security settings do not match.*

RELATED TOPICS

UNIT CONVERSIONS

OVERVIEW

Conversion classes are sets of units of measurement. Multipliers between the units in a class create conversions from one unit type to another. Conversion classes can be created so that units and conversion factors can be associated with numeric attributes. Conversion classes can only be associated with numeric attributes. When conversion classes are associated with attributes, buyers and product fullaccessusers are able to indicate which units they are using when setting attribute values.

Example: if a ”temperature” numeric attribute gets created, and there is a temperature conversion class, the attribute can be linked to the conversion class. Users, who set a temperature value, can also select whether the temperature is in Celsius, Fahrenheit or Kelvin, as long as each of those units are included in the class.

Creating Conversion Classes Admin Page Abilities

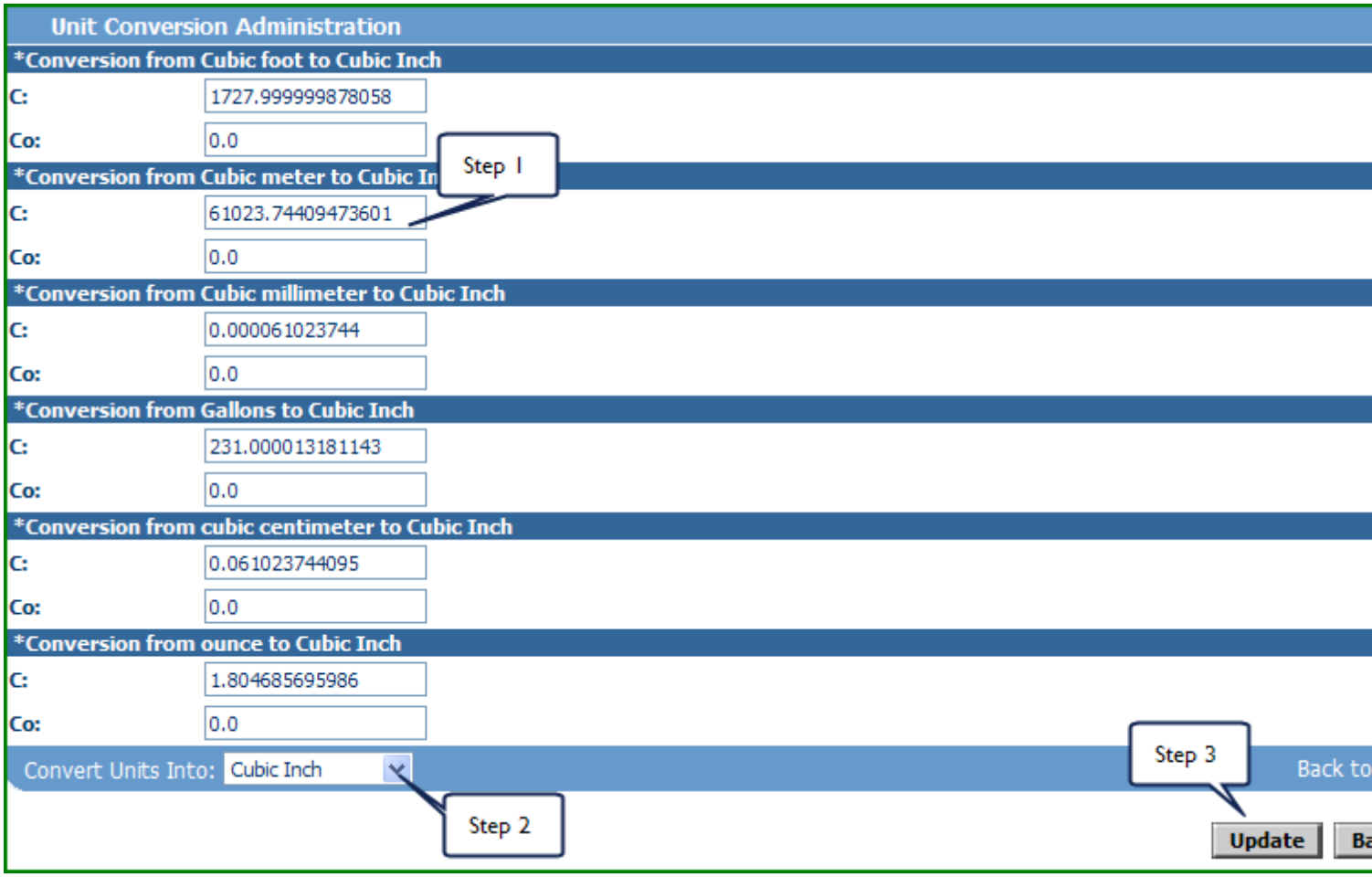
- Select a Conversion Class Name for Editing Conversion Class Names or for deleting it.
- Click the Units list link to display the page for managing units lists for the selected class name.
- View the Status of the conversion class.
- Click Add to Add Conversion Classes.
- To delete a conversion class: click the Delete column checkbox and the Delete button. Only conversion classes that have not been deployed can be deleted. *It is not possible to undo deletions, and upon a deletion, the commerce system also deletes all units and conversions contained in the class.*
- Click Deploy to deploy the new and undeployed conversion classes. Upon clicking Deploy, all pending conversion classes are deployed. Pending conversion classes are indicated with a New label on the Conversion Class List page. *Conversion classes must be deployed to be active in the commerce system. It is recommended to schedule conversion class deploys during low usage times.*
- Click Back to return to the previous page.
- Click Translations displays a window where you can enter text language translations for supported languages. *Click Save to save translations in the system and continue entering translations. Click Save and Close to save entered translations and close dialog. Click Close to close dialog without saving.*

ADMINISTRATION

Defining Unit Conversions

Create conversions by setting the multiplication factor that makes one unit of measurement equivalent to another. Conversions can be edited at any time. *When creating conversions, Co is the constant and C is the multiplication factor.*

1. Enter conversion values in the **C** fields.
2. Change which unit is the constant by choosing a different unit from the **Convert Units Into** drop-down menu. *This immediately reflects a change in the values to be converted.*



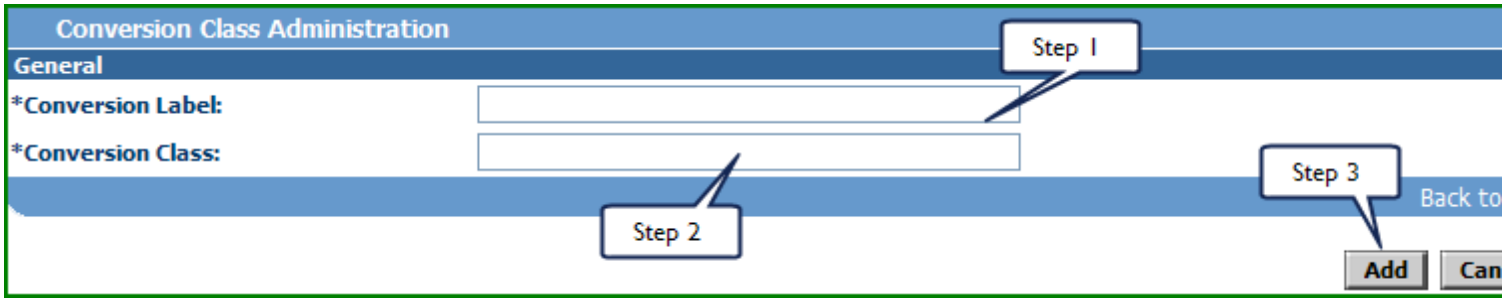
3. Click **Update**.



The default unit type specified on the Units List page is also used as the default constant. The constant (Co) should remain 0.0, except in temperature conversions. *In deployed conversion classes, conversion changes take effect immediately after clicking Update.*

Adding Conversion Classes

1. Enter a **Conversion Label**. *This is the name that identifies the class in attribute administration areas and is displayed on the Conversion Class List page.*



2. Enter a **Conversion Class**. *This is the name of the conversion class.*
3. Click **Add**. *This takes you back to the Conversion Class List page displaying the newly added conversion class.*

Conversion classes can be created one at a time.

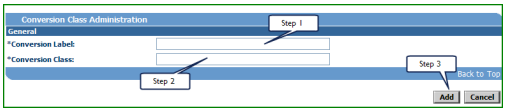
Editing Conversion Classes

1. Modify the **Conversion Label**. *This is the name that identifies the class in attribute administration areas and is displayed on the Conversion Class List page.*
2. Click **Update**. *This takes you back to the Conversion Class List page displaying the modified conversion class.*

Conversion class names can only be edited in classes that have not been deployed.

Adding a Unit

1. Enter a **Unit Label**. *This is the name that will be displayed on the Units List For* `<name of the selected Conversion Class>` *page.*



2. Enter a **Unit Class**.
3. Click **Add**. *This takes you back to the Units List For* `<name of the selected Conversion Class>` *page displaying the newly added conversion class.*

Editing a Unit

1. Click the **Units** link next to the **Conversion Class** you want to edit.
2. Click the **Unit Name**.
3. Modify the **Unit Class Label**.
4. Click **Update**.

NOTES



You can set English and metric default units for each conversion class. Whenever a conversion class is associated with a numerica attribute, the default units appear selected in the units menu. Users can override default units by picking other units in the class.



Conversions happen automatically when units switch from one to another in the same class.

Changes to default unit settings take effect immediately in deployed conversion classes.

Conversion classes are created in a non-active, pending state. After a conversion class is complete, meaning it has both units and conversions, the class can be deployed to make it active.



Only full access fullaccessusers can administer conversion classes.

RELATED TOPICS

BIGMACHINES FEATURES BY VERSION

OVERVIEW

Features are listed per release:

2013 Autumn

2013 Autumn Features

- Migration Rollback & Enhancements
  - Rollback: Undo changes on target site that was previously migrated
  - Snapshots
  - Data Columns can be deleted in migration
  - Description field for each migration
  - Improved error messages regarding BML compilation
- Formula Management
  - Improved response for quotes w/ large # of line items & sets of formulas
  - Select quote-level formulas executed when line items are added to quote
  - Advanced Default- Before Formulas
  - Advanced Default- After Formulas
  - Auto-Update Before Formulas
- Function to Function Calls
  - Commerce Library functions can call other Commerce Library functions
  - Util Library functions can call other Util Library functions
- Mobile
  - Mobile Configuration
  - Mobile Home Page
- Help System
- Secure File Manager
  - Set folders as secure
  - Improved migration capabilities for File Manager references
- Line Item Performance Enhancements
- Oracle Fusion Integration
- Translations Bundle (French, German, Spanish)
- Additional Enhancements
  - Login allows periods
  - Price Book columns increased to accommodate more currencies
  - Mass Update function can be performed on step subset
  - BMQL: Return Part Description in specified language translation
  - Reconfigure in a different language
  - BML function keys() added
  - "Web Services Only" user permissions checkbox
  - Export Attachment action language pop-up
  - Enhanced Security
  - Third Party Software
- **Unplugged Discontinued**

2013 Spring

2013 Spring Features

- Commerce
  - Email Template Tables
    - Add Tables to Email Template body
    - Drag & drop Tables into Email Template
    - Add/remove/edit Cell, Row, Column, & Table content & formatting
- Data Tables
  - Delta Data Table Uploads (Natural Keys)
    - Supported in: Data Tables, Web Services API, CSV & Bulk Upload
- Internationalization
  - Currency Enhancements
    - Enable any/all BMI-supported currencies

2013 Spring Features

        - Admin side hover tooltip displays full name of currency code
- User Management Enhancements
    - Admins informed of remaining user licenses available for use
    - Icons warn users when at or approaching user capacity

2013 Winter

2013 Winter Features

- Approval Sequences
    - Create connections for parallel to serial Reasons & Approvers
    - Pending Approval Notification reminder email can be sent to Approvers
- Email Templates
    - Translations for sites w/ multiple languages
    - Preview a completed email template
- Formula Management
    - In-line editing
    - Auto-Update on formulas thru AJAX
    - Advanced Modify- Before Formulas option on Modify actions
- Increased Decimal Places
    - Hover tooltip appears if stored precision differs from displayed precision & numbers aren't equivalent

---

Version 12.2

Version 12.2 Features

- Customer Specific Pricing
    - Defined on part-by-part basis
    - Create Profiles (groups of parts) to attach to Pricing Rules
- Configuration
    - Simple Config Rules for Arrays
        - Use with simple Recommendation, Hiding, Constraint Rule conditions
        - 1/more Arrays can be constrained
        - Constrain, Hide, Recommend Array values index-by-index
    - Table Based Rules w/ Arrays
        - Access Table data w/ point/click editor
        - Config Arrays used w/in conditions for Recommendation
- Commerce
    - Email Templates
        - Only for Pending Approval notifications
    - Formula Management
        - Set Commerce Attribute values using drag & drop editor
    - Quote conflict alert
    - Auto-Forwarding Rules may be sent to individual users
    - Refresh action added to documents
    - Revise action
- Web Services & Integration
    - SOAP API for Groups
- Data Tables
    - Data Table Validation
        - Map columns to menu attributes
        - Error messages denoting invalid data
- Increased Decimal Places
    - Host Company level: set decimal precision for different currencies
    - Individual currency attributes can have own decimal precision
    - Currency fields can support more than 2 decimal places
- General Enhancements
    - Sales users cannot access Groups
    - Data Cube Export: new field that defines # of days for which history will be downloaded

Version 12.2 Features

& Constraint Rules
    - Set/constrain Array values index-by-index
  - On-Demand Tab Loading
  - Simple Attribute comparison available in Rule conditions
  - Users prompted when Product Family has been updated
  - Multiple Configurable Attributes may have same name
  - "Recreate buy-side cache after deployment" setting added
  - Commerce SOAP API addToTransaction enhanced
  - Return specific Attributes in SOAP APIs
  - Real-Time Integration enhancements
- Document Engine
  - Generate documents as DocX output

Version 12.1

Version 12.1 Features

- Table Based Rules for Recommended Item Rules
  - Define Rec Item Rules that access Data Tables thru point/click Editor
  - Recommended Item Splitting
- Boolean Attributes
  - Can define any 2 choice option (e.g. True/False, Valid/Invalid, Include/Do Not Include)
- Pending Configurations
  - Save/Retrieve incomplete Configurations if accidentally logged out of BMI session
- Recommended Item Splitting
  - Separate multiples of same Part Number returned by a Rec Item Rule
- Approval Skip Nodes
  - Can select Reasons/Approvers w/in Approval Sequence; Non-Required Reasons/Approvers skipped if Condition = False
- My Approval Search
  - Search for Your Pending Approvals thru Transaction Manager
- Access Reasons Status in Simple Condition Grid
  - Available for Reason & Approver
- Localization
  - Managing Your Translations Through Bulk Upload & Download in preparation for International Rollout
- Data Table Audit (default = not activated)
  - Use Bulk Downloads to verify changes made to Data Tables
- Migration Management Enhancements
  - Migrate Data Tables & File Manager data
  - Verify low-level differences thru Migration Management Center
- Document Engine Validations
  - Send feedback to Dev regarding what steps were performed when errors occur
- Web Services Integration
  - Read, Attach, Delete file attachments To/From Quotes thru new Commerce SOAP APIs

---

Version 12.0

Version 12.0 Features

- Migration Management
  - One-Touch Migration
    - Commerce Process
    - Documents
    - Product Family
- Enhanced End User Pages
  - Label, Field, Message, Image, & Price position
  - Enhanced Array functionality
    - Array widths
    - Array messages
- General Enhancements
  - Config contains Operator available for User Group in Rules
  - CRM On-Demand Integration supports Stateless Web Services
  - Quantity Column size increased in Shop-

Version 12.0 Features

        - Catalog
        - Configuration
        - Util Libraries
    - View Source & Target site differences prior to Migration
    - View code change difference prior to Migration
    - Versioning & Rollback
    - Status & Logs
- New Config Layout
    - Drag & drop Editor
        - Layout Elements
        - Attribute & Spacer size
        - Admin Tools
        - AJAX Rules
        - Enhanced look/feel
        - Advanced Templates
- Commerce Rules
    - Hide Attribute using AJAX Hiding Rules
    - Constrain Attribute Values using AJAX Constraint Rules
    - New Validation Rules
- One-Page Steps Administration
    - Admin Workflow Steps, Profiles, Transition Rules, Timers
    - Access Step Variable Name quickly for easy Debugging

ping Cart & Favorites
- External SSO ID & Login Field size increased

Version 11.1

Version 11.1 Features

- Configuration
    - AJAX Configuration Rules
    - Price Book enhancements
    - Transaction ID as a Rule Input
    - New Standard Templates
- Commerce
    - Drag & drop line items
    - Group Sequencing
    - Doc Engine Conditional Columns
    - Doc Engine Migration Management
    - Delegated Approvals
    - Encrypted Commerce Attributes
    - _system_selected_document_number available for all actions
    - Approval comment available in Approval History Attribute
- Platform
    - BML enhancements
    - BMQL Transaction
    - MSCRM Online integration
- Architecture
    - JDK 1.6
    - Security enhancements
- Admin User Browser
    - Firefox 3.5.x
    - IE7 &IE8
- End User Browser
    - Firefox 3.5.x
    - IE7 & IE8
    - Safari 5.x
    - Chrome 8.x
    - iPad w/ most end user features
- General Enhancements
    - Force.com Configurator
    - Oracle Fusion Configurator
    - MSCRM Integration
    - SAPVC Integration
    - Vizualization
    - CAD Integration
    - Self-guided web-based training
    - Advanced instructor-led training
    - Advanced Admin certifications
    - UI/UX training & consultations
    - Additional Vertical Quick-Starts
    - Additional Page Templates
    - Other PS implementation tools

Version 11.0

Version 11.0 Features

- Parts Search
- Enhanced Data Tables
  - One page Admin
  - Excel look & feel
  - Powerful Search
  - Enhanced performance
- Configuration Table Lookup Wizard
- Configuration Currency Attributes
- Configuration Templates
- Quick key item entry
- Select Currency in Transactions
- Native Reporting 2.0
  - Drag & drop builder
  - Advanced Filter capabilities
  - Calculated Columns
  - Grouping by Dates
  - Report on Configurable Attributes
  - Scheduler
- Security
- Oracle 11g
- Admin Home Page update
- Product page update
- File Manager Search
- Unplugged Enterprise licensing
- Mass Update of Transactions
- Lint item pagination

Version 10.1

Version 10.1 Features

- Partner app SSO w/ SAML
- eCommerce self-registration
- Enhanced Data Tables
  - Upload
  - Deploy
  - CSV Scheduler
- Configuration Data Attributes
- Configuration Image Menus
- Enhanced Constraints
- Unplugged
  - Support 32k+ Parts, Accounts, etc.
  - Improved Admin UI
  - Support multiple JDK
- Sticky action in Config
- BMQL or BQOOL
- Collapsible Header

Version 10.0

Version 10.0 Features
• HTML Attributes • New User & Admin interface • Standard Approvals • Document Engine Condition Library • Shopping Carts • Unplugged enhancements • New Commerce page layouts • eCommerce Shopping Cart • Approval Sequences • Technology enhancements

---

Version 9.2

Version 9.2 Features
• Pricing Options • Mobile Approvals • Data Migration • Configuration & Pricing • Admin tracking • Config Attribute Pricing • Mobile Device/Email Approvals • Commerce Source-to-Target Migration • AJAX field updates • Admin Change Log & Control • CRM user integration

---

Version 9.0

Version 9.0 Features

• Document Engine
• RTE (rich text) Fields & Templates
• BML Libraries
• Simple Rules
• Configuration Rule Editor
• BML Function Editor
• Accounts CSV
• Superuser access to error logs
• Superuser to proxy as Partner Organization supervisor
• Generate XSL proposals similar to PDF/RTF
• Reporting
  ○ Search on multiple values in a menu
  ○ Increased number of columns on quote & line item documents

NOTES



As of 2013 Autumn, Unplugged will no longer be supported.

RELATED TOPICS

BROWSER COMPATIBILITY

OVERVIEW

The following lists browser compatibility by BigMachines version:

2013 Autumn

Desktop:

- *User-Side:* IE8, IE9, IE10, FF20.x, and Chrome 26.x
- *Admin-Side:* IE8 and FF20.x

Mobile:

- *iOS 6:* Safari 6.x
- *Android 4.1 Jelly Bean:* Chrome 26.x

2013 Spring

- **User-Side:** IE8, IE9, IE10, FF19.x, Safari 6.x, and Chrome 24.x
- **Admin-Side:** IE8 and FF19.x

2013 Winter

- **User-Side:** IE8, IE9, FF16, FF17, Safari 5.x, Safari 6.x, and Chrome 23.x
- **Admin-Side:** IE8, FF16, and FF17

Version 12.2

- **User-Side:** IE8, IE9, FF7, Safari 5.x, and Chrome 12.x
- **Admin-Side:** IE8 and FF7

Version 12.1

- **User-Side:** IE8, IE9, FF7, Safari 5.x, and Chrome 12.x
- **Admin-Side:** IE8 and FF7

Version 12.0

- **User-Side:** IE7, IE8, IE9, FF7, Safari 5.x, and Chrome 12.x
- **Admin-Side:** IE8 and FF7

Version 11.1

- **User-Side:** IE7, IE8, FF4, Safari 5.x, and Chrome 12.x
- **Admin-Side:** IE7, IE8, and FF4

Version 11.0

- **User-Side:** IE7, IE8, FF3.5.x, Safari 5.x, Chrome 8.x
- **Admin-Side:** IE7, IE8, and FF3.5.x

Version 10.1

- IE7, IE8, FF3.6

Version 10.0

- IE7, IE8, FF3.5

Version 9.2

- IE6, IE7, FF3.5

Version 9.0

- IE6, IE7, FF3.0

NOTES



For more information regarding browser settings, see Recommended Browser Settings.



Functionality can not be guaranteed on any other versions. Before submitting a case, please verify that you and/or your users who have experienced a problem are using a supported browser and version.



Sites created with additional JavaScript, alternate CSS, or other custom functionality, may no longer be compatible with our supported browsers. Please contact Customer Support to determine if work-arounds and minor fixes are available. Add-on work may be required to support customizations.

RELATED TOPICS

THIRD PARTY SOFTWARE

OVERVIEW

The following lists the copyright information for third party software used in the BigMachines application.

ADMINISTRATION

Copyright Information

Antenna House XSL Formatter

- Antenna House XSL Formatter 5 Copyright (C) 2000-2010 Antenna House, Inc. All rights reserved.

Colorbox

- jQuery ColorBox v1.3.30 (c) 2013 Jack Moore - jacklmoore.com/colorbox updated: 2013-01-25

Deja Vu Fonts

- Bitstream Vera Fonts Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitsream, Inc.

Handlebars

- Handlebars Copyright © Yehuda Katz

Inconspicuous

- JSMin.java 2006-02-13 Updated 2007-08-20 with updates from jsmin.c (2007-05-22) Copyright (c) 2006 John Reilly (www.in-conspicuous.org)
- jsmin.c 2003-04-21 Copyright (c) 2002 Douglas Crockford (www.crockford.com)

JavaScript Toolbox

- JavaScript Toolbox Copyright (c)2005-2009 Matt Kruse (javascripttoolbox.com)

JCProv

- ProtectToolkit C © 2009 SafeNet, Inc. All rights reserved.

---

jsdifflib

- jsdifflib Copyright (c) 2007, Snowtide Informatics Systems, Inc. All rights reserved.

JSON Library

- JSON Library Copyright (c) 2002 JSON.org

JSPBook

- JSPBook Copyright (c) 2002-2004 All rights reserved. Copyright 2003 Jayson Falkner (jayson@jspinsider.com)

Jython: Python for the Java Platform

- Jython Copyright (c) 2007 Python Software Foundation; All Rights Reserved.

Microsoft Fonts

- Arial Unicode Digitized data copyright (C) 1993-2000 Agfa Monotype Corporation. All rights reserved. Arial™ is a trademark of The Monotype Corporation which may be registered in certain jurisdictions.
- MS Gothic (C)2006 data:RICOH Co.,Ltd. typeface:RYOBI IMAGIX CO. MS Gothic is a registered trademark of the Microsoft Corporation.
- MingLiU and PMingLiu (c) Copyright DynaComware Corp. 2008 Trademark by DynaComware Corp.
- SimHei © Beijing ZhongYi Electronics Co., 1995-2005, All rights reserved. Trademark of Beijing ZhongYi Electronics Co., China
- SimSun © Copyright ZHONGYI Electronic Co. 2001 Trademark of ZHONGYI Electronic Co., Beijing
- Verdana © 2012 Microsoft Corporation. All rights reserved. Verdana is a trademark of the Microsoft group of companies.

QUnitJS

- QUnitJS Copyright 2013 jQuery Foundation and other contributors http://jquery.com/

TinyMCE

- TinyMCE Copyright 2009, Moxiecode Systems AB Released under LGPL License.

XFC (XMLmind XML-FO Convertor)

- XMLmind XSL-FO Converter Copyright © 2002-2009 Pixware SARL

Yahoo UI Extensions Library

- YUI Copyright © 2013 Yahoo! Inc. All rights reserved.

RELATED TOPICS

MANUALLY ADDING A TABLE

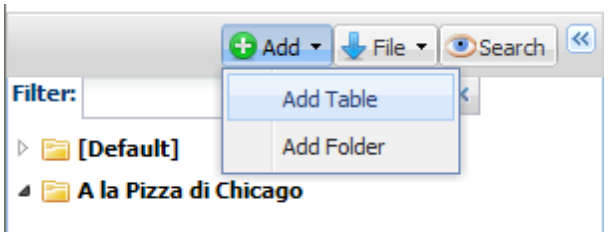
OVERVIEW

There are two different ways to add Data Tables to BigMachines: manual entry or through an import. In this section, we will discuss how to manually add a table, its schema and its data.

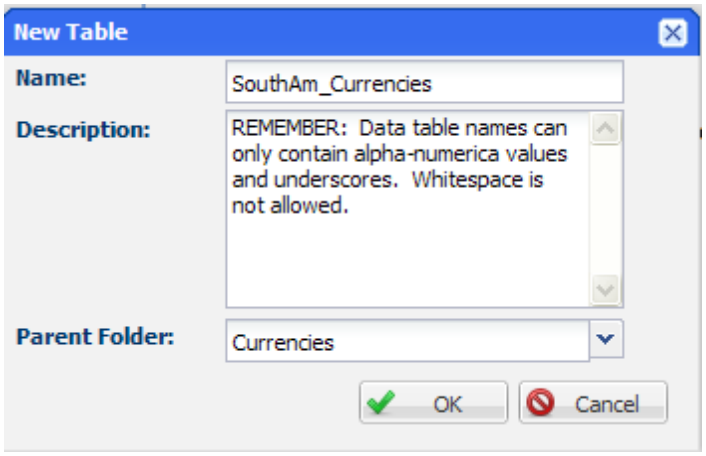
ADMINISTRATION

Adding a Table

1. From the Admin Toolbar, click **Add** and choose **Add Table** from the drop-down menu.



2. Enter the table **Name** and a **Description**.





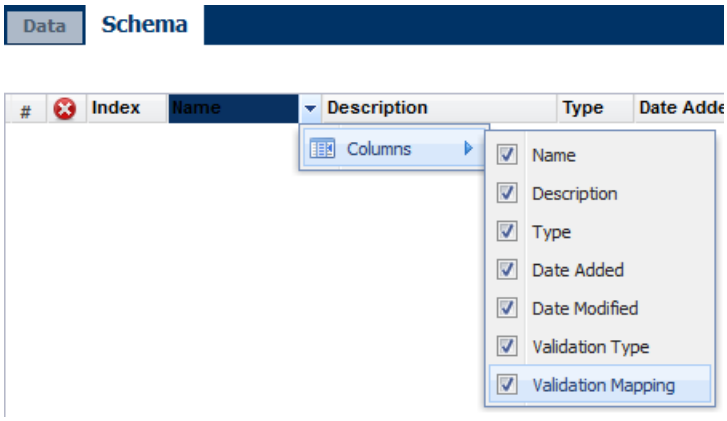
You can also select which folder it should be placed in from the Parent Folder and Drop-Down menu. Data table names can only contain alpha-numeric values and underscores. White space is not accepted.

3. Click **OK**. The new data table will now be displayed in the Folder List beneath the selected Parent Folder.

Adding a Data Table Schema

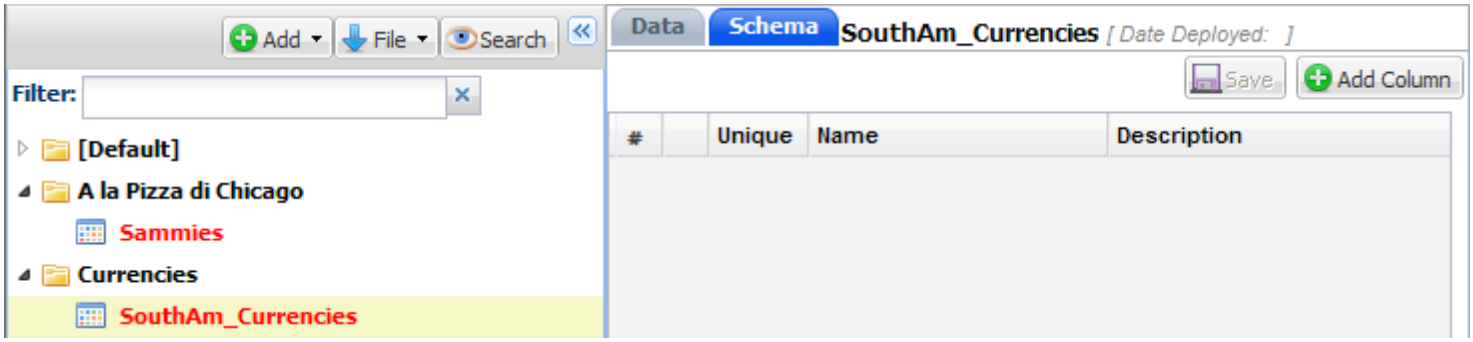
You will notice that from the Data Table Admin page, the right panel will now display the title of the table, but there isn't a way to add information. Before you can add data, you will need to set the schema of the table.

1. Click on the **Schema** tab in the right panel of the **Data Table Admin** page.
2. Click on the schema header **Name** and click on the menu option **Columns**.



From here, you can select which headers you would like to be displayed when you are looking at data table schema: Name, Description, Type, Date Added, Date Modified, Validation Type and Validation Mapping.

3. Click the **Add Column** button to add a new data table column.



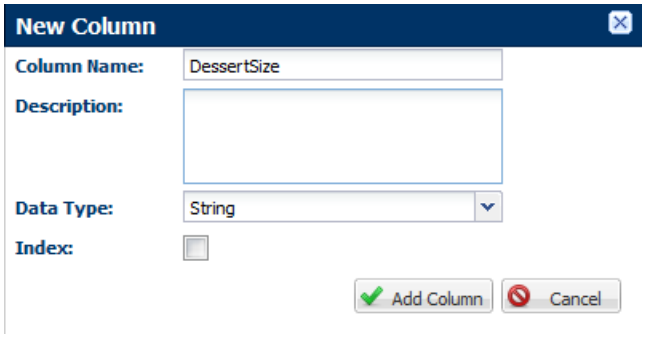
4. Enter a **Column Name** and **Description**.

Column names can only contain alpha-numeric characters and underscores. White space is not allowed.

5. Select the **Data Type** of the column: String, Integer or Float
6. Select the **Index** checkbox if you will be using BMQL statements.

> By selecting an index, performance will be improved when you are using the column in a "where" clause.

7. Click **Add Column** to save the column and return to the **Data Tables Admin** page.



8. Repeat **Steps 2-7** for each column in your data table.

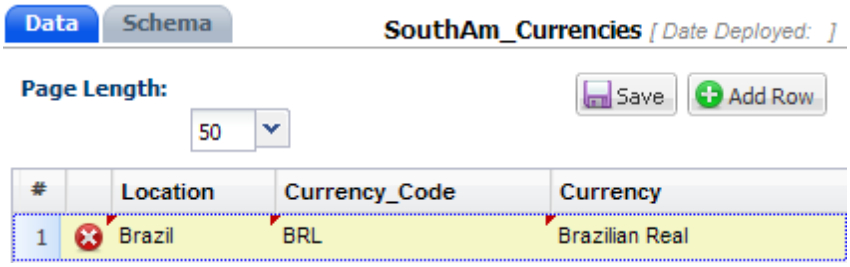
Populating the Data Table

Once you have set the schema for your table, you can begin to add the data.

1. Click on the **Data** tab in the right panel of the **Data Table Admin** page.

> The columns you created/imported in the **Schema** tab now appear on the data table.

2. Click the **Add Row** button.
3. Enter values for each column. *You can either click into each field or you can use Tab to move quickly through the fields.*

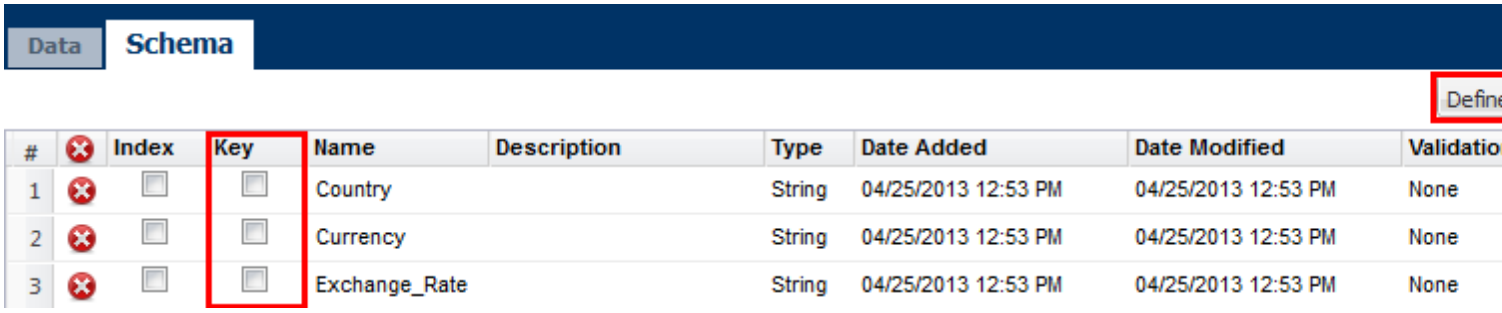


4. Repeat **Steps 2-3** until your table is complete.
5. Click **Save** once you have populated your table.

Adding Natural Keys (Delta Data Table Uploads)

BigMachines has included Natural Keys within data tables to improve efficiency. As a result of Natural Keys, the bulk upload files that modify data table data now only require the table records that are to be changed. Natural Key can be a single column or a combination of several columns that produce a unique identifier for each record (row). The creation of these Natural Keys reduces the maintenance downtime of data tables by including only data table changes upon upload. Users no longer need to purge and reload all data table rows to edit a subset of rows. For more information, please see Delta Data Table Uploads (Natural Keys) article.

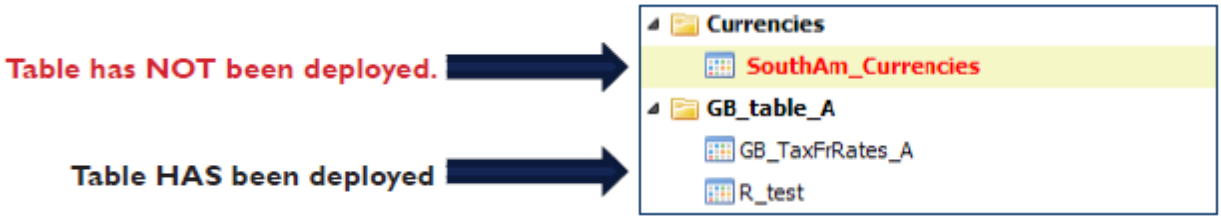
1. Navigate to the **Schema** tab.
2. Locate the **Natural Key** column.



3. Check the **Key** boxes to identify which Natural Keys will be unique identifiers OR
4. Use the **Define Keys button** to select your Natural Keys.

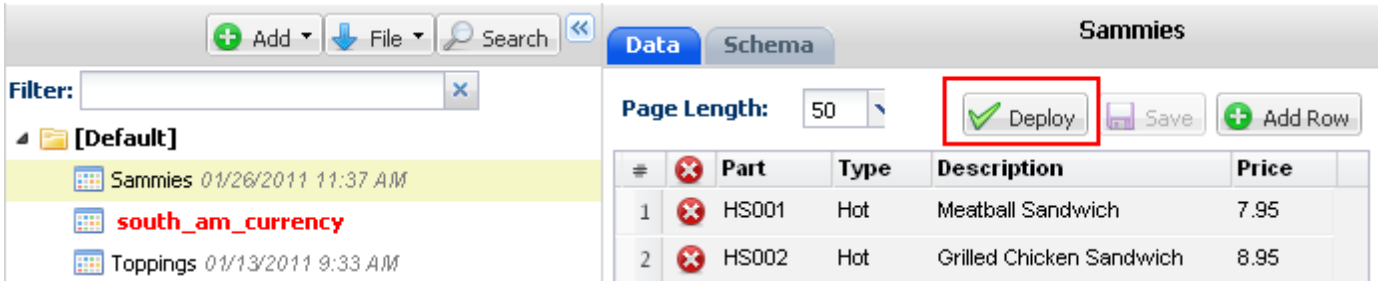
Deploying a Data Table

The final step to manually adding a table is to make sure it is deployed. All tables that have been added or modified and are NOT deployed will be clearly marked in red.

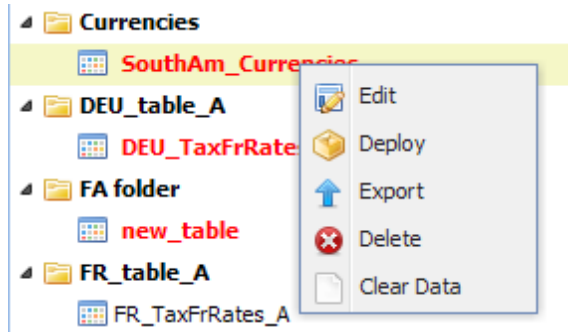


Do one of the following:

1. Click the **Deploy** button when you have a data table open.



2. Right-click on data table name. This will display multiple options for the table:



Edit: This allows you to edit the table description and view the Date Added, Date Modified and Date Deployed time stamps.

Deploy: Deploying saves the table and makes it available for use throughout BigMachines.

Export: Exports the data table to a .csv file.

Delete: Deletes the table, including its data and schema.

Clear Data: Deletes ALL table data but will keep the table and schema intact.

NOTES

To quickly populate tables, add multiple rows BEFORE entering data values.

RELATED TOPICS

DATA TABLE OVERVIEW

OVERVIEW

The single-page administration includes spreadsheet-like user interface with easy editing capabilities, powerful search features that allow searches within a single table or across all tables, and enhanced performance and indexing.

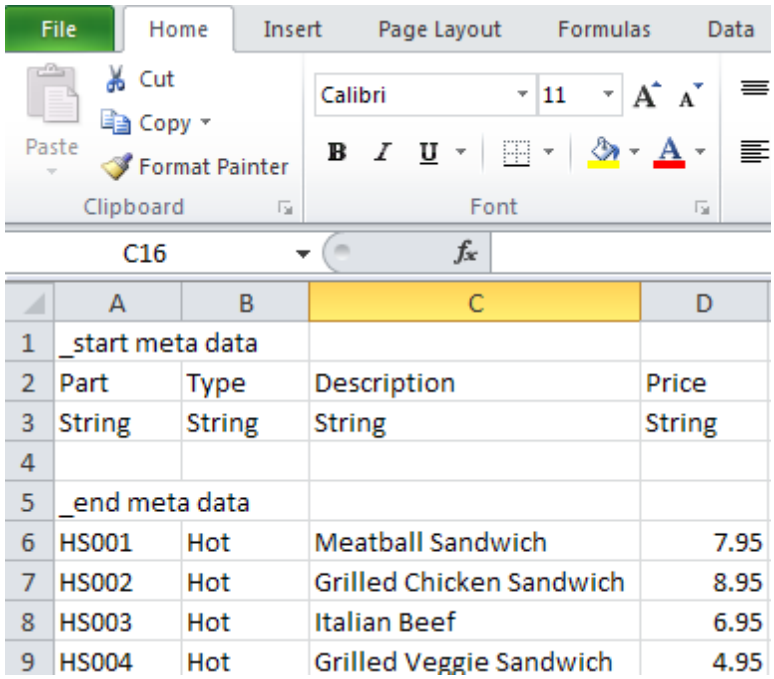
Data Table limits

ITEM	LIMITS
Total Columns in a Table	100 Columns (50 String, 25 Integer, 25 Float)
Total Rows in a Table	1,000,000 + Rows
String Characters in a Cell	256 Characters
Integer Characters in a Cell	16 Characters
Float Characters in a Cell	9 Characters

Data Table storage

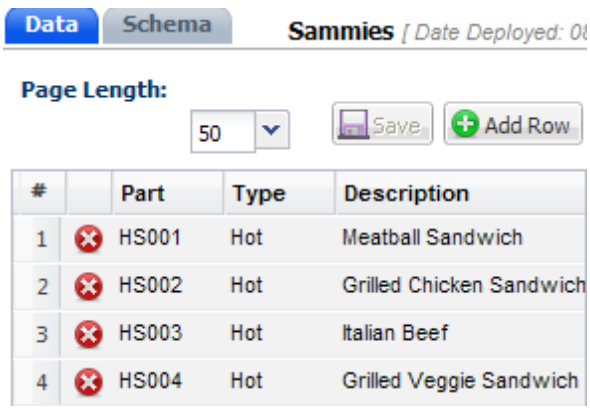
Data Tables are used to store product and commerce data. They can be accessed from any part of the BigMachines application where BML is used. The purpose of storing complex business logic in data tables is for ease of maintenance.

Data Stored in Excel (CSV)



	A	B	C	D
1	_start meta data			
2	Part	Type	Description	Price
3	String	String	String	String
4				
5	_end meta data			
6	HS001	Hot	Meatball Sandwich	7.95
7	HS002	Hot	Grilled Chicken Sandwich	8.95
8	HS003	Hot	Italian Beef	6.95
9	HS004	Hot	Grilled Veggie Sandwich	4.95

Data Stored in BigMachines



#		Part	Type	Description
1		HS001	Hot	Meatball Sandwich
2		HS002	Hot	Grilled Chicken Sandwich
3		HS003	Hot	Italian Beef
4		HS004	Hot	Grilled Veggie Sandwich

Migration

When migrating configuration or commerce, a list of dependent data tables will be displayed in a pop-up window. If these data tables are not selected, migration will fail. This pop-up will only appear when menu attribute validations are mapped to a data table, but not when the data table is used in BMQL rules.



If you delete an attribute or menu value when migration changes attribute menu values, the link will be removed from the data table.

Natural Keys

BigMachines has included Natural Keys within data tables to improve efficiency. As a result of Natural Keys, the bulk upload files that modify data table data now only require the table records that are to be changed. Natural Key can be a single column or a combination of several columns that produce a unique identifier for each record (row). The creation of these Natural Keys reduces the maintenance downtime of data tables by including only data table changes upon upload. Users no longer need to purge and reload all data table rows to edit a subset of rows.



When using the Data Table Natural Keys feature, it is strongly recommended that no more than five string columns be assigned to a part of the key.



For more information regarding Natural Keys, please see Natural Keys (Delta Data Table Uploads).

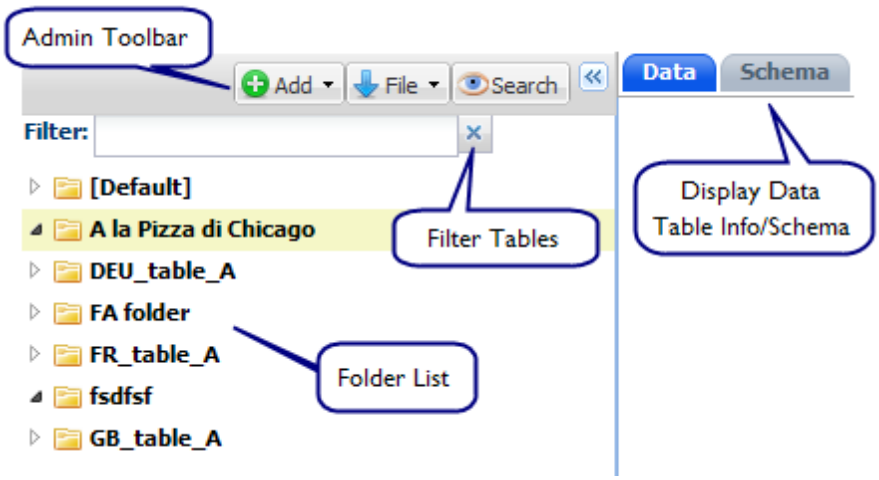
ADMINISTRATION

Accessing Data Tables

Information in data tables can be accessed from anywhere in your BigMachines application that uses advanced functions. To manage data tables, navigate to:

1. Top or Side Navigation Menu **OR**
2. Data Tables section on the Admin Home Page

After clicking a Data Tables link, you will be taken to the Data Table Administration page. From this page, you will be able to add tables and folders, import and export table files, and perform searches across tables.



TROUBLESHOOTING

If an offline back-up is still needed

1. Log a case requesting a retrieval of information as of XX date/time. **NOTE:** BigMachines only creates back-ups of the entire site. If you require specific data table(s) back-ups, you must do so yourself.

2. We do full back-ups every Sunday evening, and incremental backups every 30 minutes. We store back-ups for a while, so previous backups are available if needed. The recovery process is labor intensive and will take many hours.

3. We will load the recovered data to your test site; this overwrites other changes that have been made.

NOTES

Make a local back-up of a changed data table by exporting before making any changes. This will provide a very quick and easy way to recover the data table in the event of errors. Make all changes to test site first.

RELATED TOPICS

SEARCHING FOR DATA WITHIN A TABLE

OVERVIEW

You can search for a particular data record within the data tables. BigMachines provides the ability to select tables to search in, columns to search on, and values to search for. Searches will return the tables that contain the data record and display only the rows of matching records within those tables. You can click on the results to be taken to the cell you need to edit.

ADMINISTRATION

Searching within a Data Table

1. Navigate from **Admin Home Page > Data Tables > Search**
2. Select the radio button next to **Search within table** and use the drop-down menu labeled **Tables** to choose the table you'd like to search for data within.

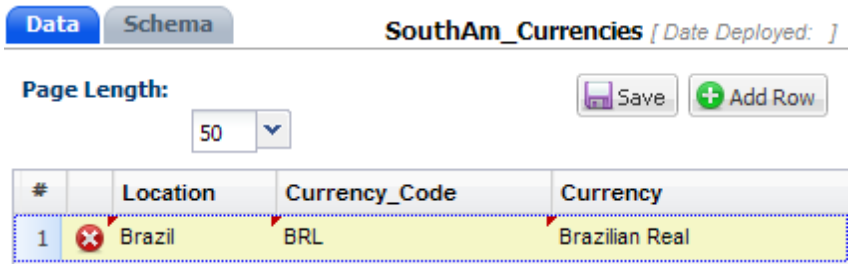
> Once you have selected your table, the schema for that table will appear, displaying column names and data types.

3. Use the drop-down menus under the **Operator** header to assign the values you would like to search for and click **Search**.



> Based on Search Results, you will be taken to the data table you selected. All rows that match your search values will be returned.

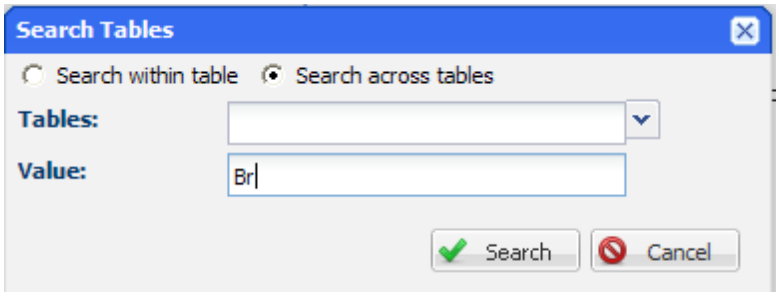
4. Click the red **X** to delete a row or click into any field to make changes.



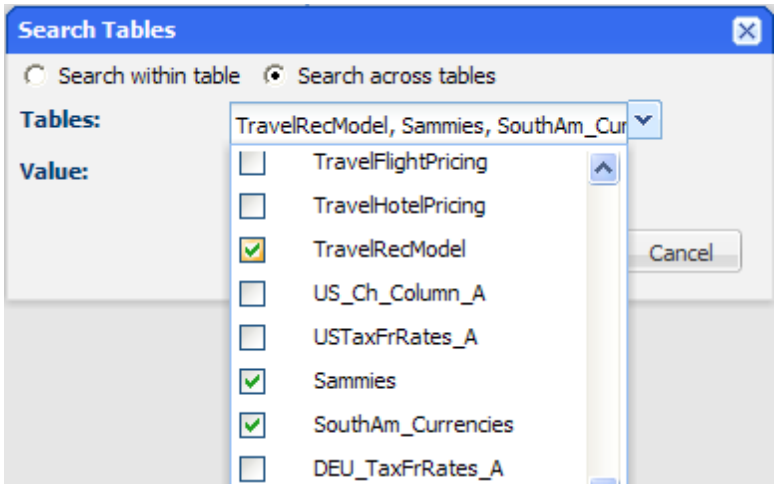
5. Make sure to click **Save** if any changes were made.
6. Click **Clear Search Results** to remove the filtered search result and display the entire table.

Searching across all Tables

1. Select the radio button next to **Search** across tables.



2. Select the checkboxes beneath the **Tables** drop-down menu to perform a search on multiple tabs.

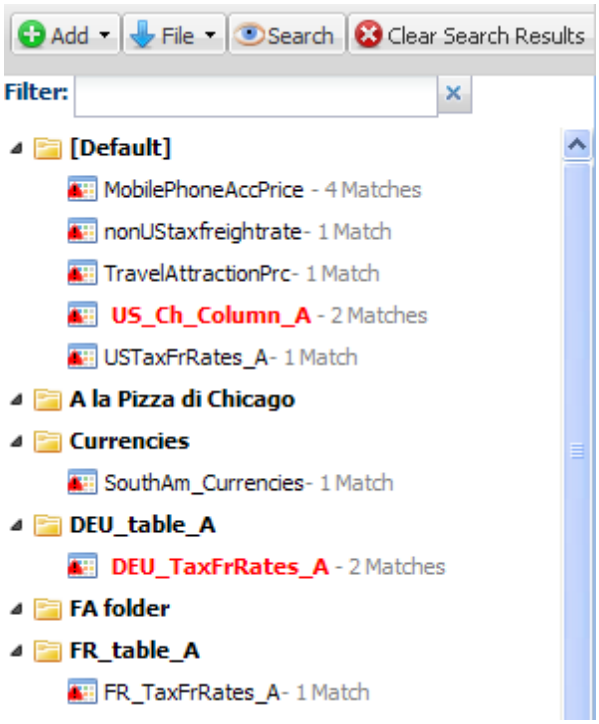




If you do not select any tables, it will search ALL tables.

3. Enter a **Value** to search for in the text field and click **Search**.

4. In the left-panel **Folder List**, the tables that have matching values will be displayed will blink red. **NOTE:** You will see the number of matches found within the table to the right of the table name.



5. Double-click the name of the table with matches. Click the red **X** to delete the row or click into any field to make changes.

> All rows that match your search values will be returned.

6. Click **Save**.

7. Repeat **Steps 5-6** for each data table.

8. Click **Clear Search Results** to remove the filtered search result and display the entire table.

RELATED TOPICS

TABLE BASED RULES

OVERVIEW

BigMachines provides a point-and-click editor to define recommendation and constraint rules that access data tables, without resorting to BMQL. The new table based-rules allow for:

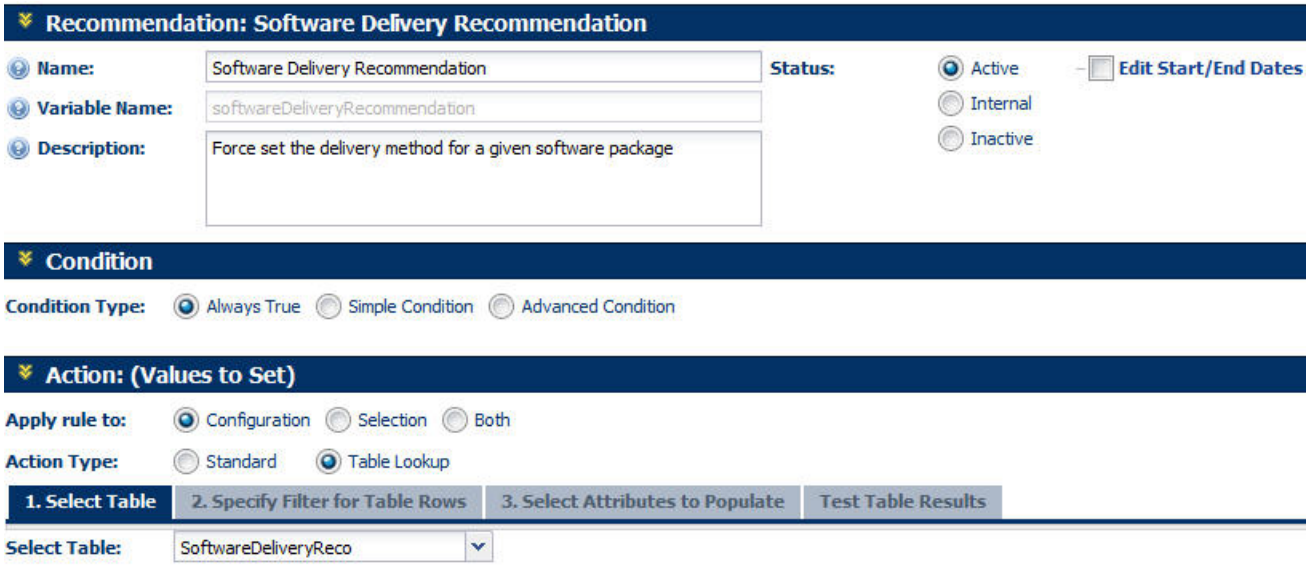
- Ability to return attribute values along with table data
- Query Debugger
- Point and Click UI for BMQL

Table based rules fire on the user-side when the condition is met. However, if no results are returned from the query then there is no impact on the action attributes. When multiple rows are returned for an action attribute, all returned values are recommended or constrained.

ADMINISTRATION

Creating a Table Based Rule

1. Using the Configuration Quick Links, navigate to Product Family > List of Recommendation Rules.
2. Click **Add** to create a new rule.
3. Enter the **Properties** of the rule.
4. Select the **Condition Type**.
5. Choose to apply the rule to either **Configuration, Selection or both**. *Configuration is the default selection.*
6. Select **Table-Based** as your Action Type. This will open a field of four tabs.



If you are creating a Recommended Item Rule, there will be an option Split Items.

7. Select the **data table** from which you would like to retrieve data, in the first tab,
8. Click the **Specify Filter for Table Rows** tab and select to Evaluate the Row with:
    - **Static Entry:** You define a static value under the Value header.
    - **Attribute:** You select a configurable attribute to map the Table Column to in a drop-down menu that appears under the Value header.



Array attributes used in the filter will execute for each array index. Also, array attributes with the same control attribute can be used in the same filter.





9. Enter the order of operations in the **Row Grouping** section.



If you only have one condition, you will still need to enter the value of 1 in this field.

10. Click into the **Value** cell and select attribute values from the drop-down menu or enter a static value. This is based on what you selected in Step 8.
11. Click on the **Select Attributes to Populate** tab.
12. Click **Add Row** to add:
    - The attribute(s) whose value you will populate with table data.
    - The table column that will populate the attribute

- Set Type (None, Set or Forced Set), and
- Message Type (Static or from a Table Column).



For Recommendation Rules: Specify the action attribute and the table column from which the recommendation value will be chosen. In addition, select a Set type for the rule.

For Constraint Rules: Specify the action attribute(s) and the constrained value. For menu-type action attributes, choose the constraint type: Constrain or Allow.

13. Click the **Test Table Results** tab.
14. Select the **Table Column and Value** you'd like to test and then **click Run**.

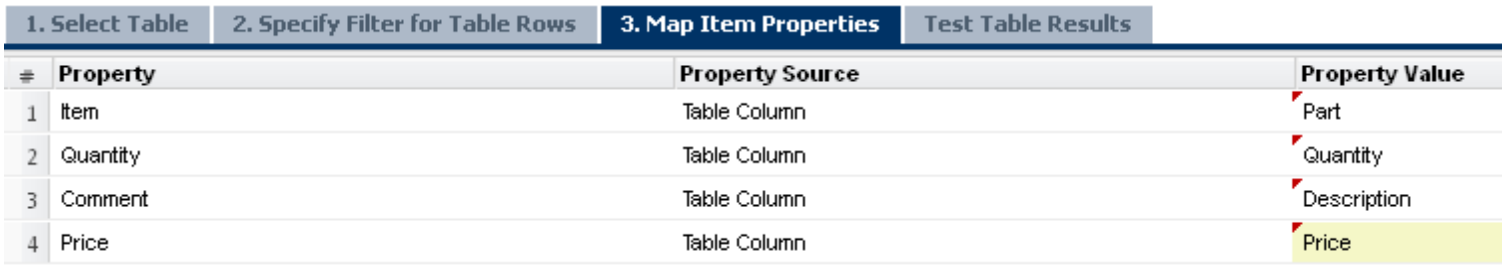


What is returned as an Ouput is the information from the table that will be returned when a user selects Meat Lovers as the Specialty Type (in this case).

Creating a Recommended Item Table Based Rule

Creating a Recommended Item Table Based rule is similar to how you'll use tables in Recommendations and Constraints. The main difference is instead of the 3rd tab being Select Attributes to populate, you'll have a Map Item Properties tab.

1. Follow **Steps 1-10** from Creating a Table Based Rule.
2. Click the **Map Item Properties tab**. Here you will map attributes, static values or table columns to the item, quantity, comment and price (i.e. return string columns).



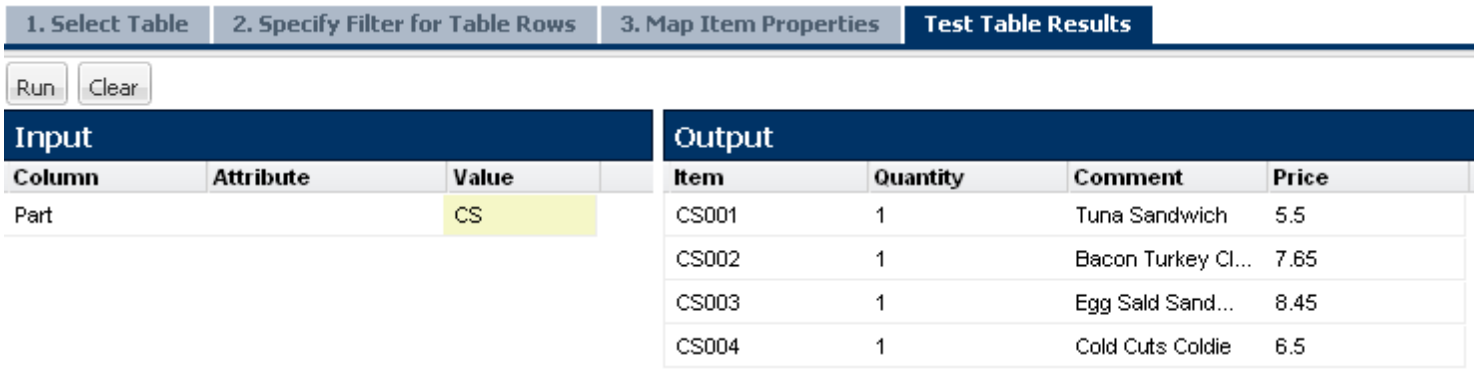
#	Property	Property Source	Property Value
1	Item	Table Column	Part
2	Quantity	Table Column	Quantity
3	Comment	Table Column	Description
4	Price	Table Column	Price

HTML and Date attributes cannot be used as inputs for Map Items Properties.

Array attributes can only be mapped to a return column when at least one array attribute with the same control was specified in the filter. The same index used in the filter will also be used to populate the mapping results

Property (Recommended Item Display Columns)	Property Source Options	Property Value
Item	Table Column	You must map directly to a table column. For example, "Part".
Quantity	Table Column Static Entry Attribute	You can map to a table column, you can define a value or you can map it to a configurable attribute. **Note: If you map the quantity field to a Table Column,the table column MUST be of an integer data type.**
Comment	Table Column Static Entry Attribute	You can map to a table column, you can define a value or you can map it to a configurable attribute.
Price	Table Column Attribute Item Master	You can map to a table column, you can map to an attribute or you can map directly to the items master (parts database).

3. Click the **Test Table Results tab**. In the Input Section, the Table Column will already be set (and read-only).
4. Enter the **Value** you'd like to test and then click **Run**.



BEHAVIOR

Action Attributes in Table Based Rules

The supported action attributes are Text, Integer, Float, Currency, HTML and Data type configurable attributes.

Attribute Type	Data Table Column Type	Supported Values	Comment
Text	String	All string values	
Integer	Integer	All integer values	
Float	Float	All float values	
Date	String	Values in database date format: yyyy-MM-dd hh:mm:ss	
HTML	String	All string values	Not supported in constraint rules.
Currency	Float	All float values	For multi-currency applications, exchange rate is applied on the values returned for the currency attribute.
Text Single Select Menu	String	All text values	
Integer Single Select Menu	Integer	All integer values	
Float Single Select Menu	Float	All float values	
Text Multi Select Menu	String	All text values	Multiple values can be recommended or constrained if '~' delimited values are returned.
Integer Multi Select Menu	Integer	All integer values	
Float Multi Select Menu	Float	All float values	

Array Attributes in Table Based Rule Actions

Recommendation and Constraint table-based rule functionality allow arrays to be used in the actions. You can use recommendation rules to set values or use constraint rules to constraint values of individual array indexes. You can use the value(s) of an array set to pull information from a data table by mapping an array attribute to a data table column. The column will be used in the filter to evaluate the array attribute value. For a single array attribute, each index will be evaluated individually against the data table.

Recommendation Rules Notes

- If there are no array attributes in the filter tab, the results will be applied to array attributes at all indexes in the result tab.
- Set types will remain the same as current recommendation rules (None, Set, Force Set).
- No match for a particular index will leave the index value and the set type 'as is'.
- The first row that matches the filter will set the array attribute index.

Constraint Rules Notes

- Constraint behavior for each array attribute will be the same as current constraint rules (Allow, Do Not Allow).
- If there are not matches from the data table, for a particular index, constraints will not be applied to that index.
- If no array attributes are in the filter that calls the data table, then all indexes of the array attribute will be constrained based off of the returned value.
- The filter will be compared against all rows in the table and all matches will constrain the related index.
- If you specify only non-array attributes in the Table Query Condition, the action will apply to all indexes of the specified array attribute.

Notes about Behavior

Simple table based rules operate per index; they will not operate across indexes. Attribute type limitations are consistent with existing functionality. The table below represents how rules will operate depending on the condition and filters.

	Non-Array Attributes Only (in mapping)	Array Only Attributes (in mapping)	Both Array & Non-Array Attributes (in mapping)
Condition: Always True **Filter**: Static Values	Map first returned row to non-array attribute.	Map first row to each index of array. Same value for every index.	Map first row to each index of array and to the Non-Array.
Condition: Always True **Filter:** Non-Array Attributes	Map first returned row to non-array attribute.	Map first row to each index of array. Same value for every index.	Map first row to each index of array and to the Non-Array.

	Non-Array Attributes Only (in mapping)	Array Only Attributes (in mapping)	Both Array & Non-Array Attributes (in mapping)
Condition: Always True **Filter:** Array Attributes	All indexes run filter; The recommended value will be the last array index value.	All indexes will run filter; Each index applies the first row of the index's filter results; Only indexes with results will perform the recommendation or constraint.	All indexes will run filter; Each index applies the first row of the index's filter results; Only indexes with results will perform the recommendation or constraint. Non-Array attribute is set to the last result returned by the last index.
Condition: Non –Array Attributes **Filter:** Static Values	Map first row returned to Non-array attribute.	Map first row to each index of array. Same value for every index	Map first row to each index of array and to the Non-Array.
Condition: Non-Array Attributes **Filter:** Non-Array Attributes	Map first row returned to Non-array attribute.	Map first row to each index of array. Same value for every index	Map first row to each index of array and to the Non-Array.
Condition: Non-Array Attributes **Filter:** Array Attributes	All indexes run filter; The recommended value will be the last array index value.	All indexes will run filter; Each index applies the first row of the index's filter results; Only indexes with results will perform the recommendation or constraint.	All indexes will run filter; Each index applies the first row of the index's filter results; Only indexes with results will perform the recommendation or constraint. Non-Array attribute is set to the last result returned by the last index.

NOTES



Only "=" and "Not=" operators are supported for querying blank values on text type attributes. If any other operator is used for blank values, the query will assume that the operator is "=" for "Contains", "Start With" and "ends With". Similarly, the query will assume the opertator is "Not=" for "Not Contains", "Not Starts" and "Not Ends with" when querying for blank values.

The user can specify multiple action attributes in the Select Attributes to Populate tab for any table-based rule. This behaves similar to standard configuration rules, which can have multiple action attributes where the rules fires per action attribute and not all action attributes have to be present in the flow for the rule to fire for one particular attribute. The rule fires if the condition is met and the condition attribute and the action attribute are in the flow.

All table data must be deployed to see the correct results in the Debugger.



Duplicate values returned by the query are removed and only unique values are recommended or constrained.

When deploying configuration, you can deploy data tables that are used in Table Based Rules, which have undeployed changes.

You can include data tables that are used in table-based rule in configuration test to production. If the data table does not exist in production, it will be uploaded to the Default folder. If the data table exists in production, then the test site data will be appended to the production site data for that table.

If multiple messages are returned, then all messages are displayed in a comma delimited list to the end user, no matter the value chosen by the user.

Array attribute type limitations are consistent with existing functionality. For example, if the table column is of numeric data type, then the array attribute(s) must be numeric.

In cases where quantity <=0, quantity will be shown as Qty = 1.

Prices will show as the following in these situations:
o On Request<=0 : Non Number value that is not the keyword “Base”
o Not DefinedNo price at all, or blank
o Default Unit PriceKeyword “Base” is found in the table



Blank queries are not supported for numeric data types.

Simple table based rules operate per index; they will not operate across indexes.

Only array attributes from the same set can be used in the Filter and Mapping tabs. Mix and match of array attributes across sets is not allowed.

RELATED TOPICS

MANUALLY ADDING A TABLE

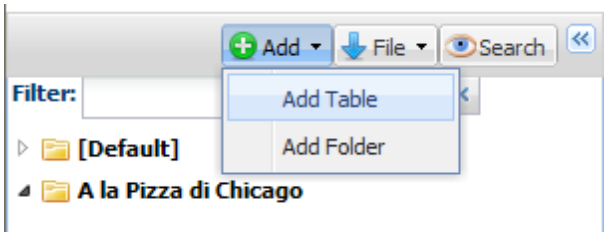
OVERVIEW

There are two different ways to add Data Tables to BigMachines: manual entry or through an import. In this section, we will discuss how to manually add a table, its schema and its data.

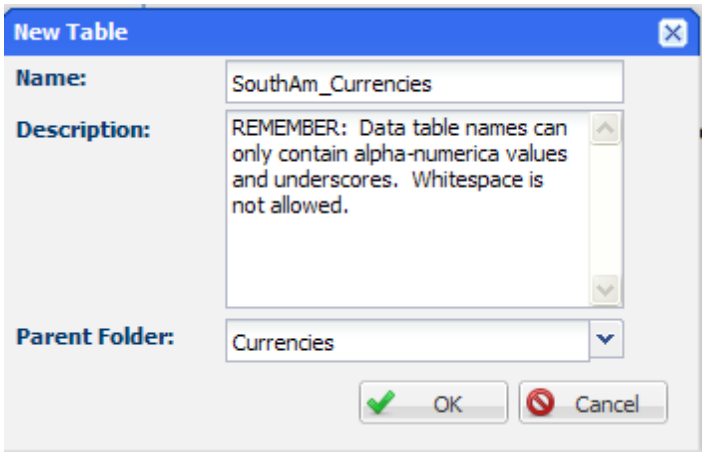
ADMINISTRATION

Adding a Table

1. From the Admin Toolbar, click **Add** and choose **Add Table** from the drop-down menu.



2. Enter the table **Name** and a **Description**.





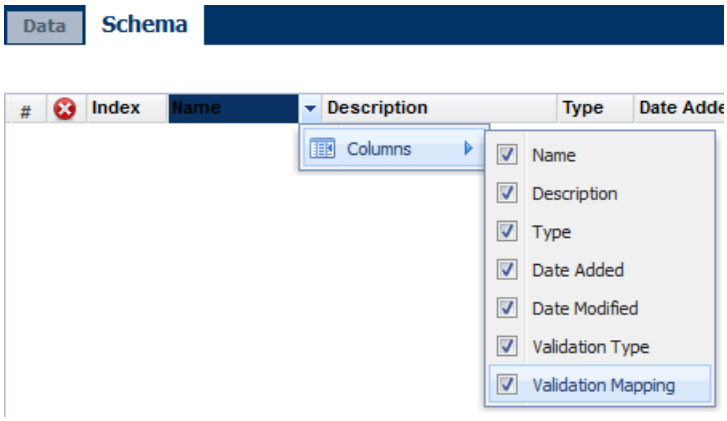
You can also select which folder it should be placed in from the Parent Folder and Drop-Down menu. Data table names can only contain alpha-numeric values and underscores. White space is not accepted.

3. Click **OK**. The new data table will now be displayed in the Folder List beneath the selected Parent Folder.

Adding a Data Table Schema

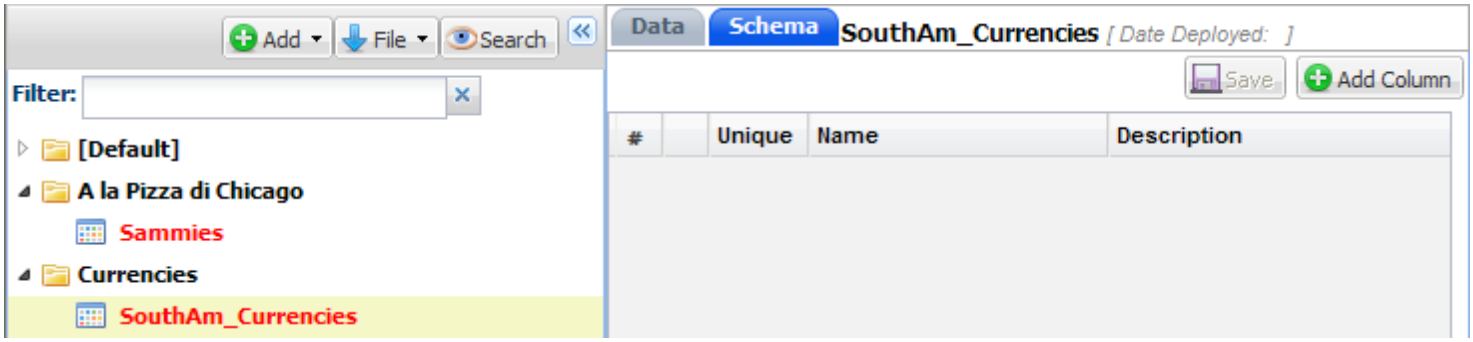
You will notice that from the Data Table Admin page, the right panel will now display the title of the table, but there isn't a way to add information. Before you can add data, you will need to set the schema of the table.

1. Click on the **Schema** tab in the right panel of the **Data Table Admin** page.
2. Click on the schema header **Name** and click on the menu option **Columns**.



From here, you can select which headers you would like to be displayed when you are looking at data table schema: Name, Description, Type, Date Added, Date Modified, Validation Type and Validation Mapping.

3. Click the **Add Column** button to add a new data table column.



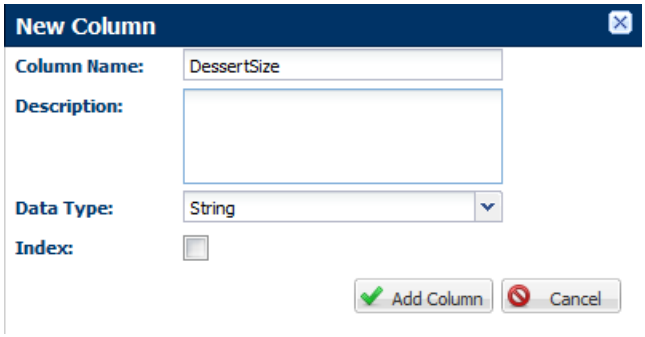
4. Enter a **Column Name** and **Description**.

Column names can only contain alpha-numeric characters and underscores. White space is not allowed.

5. Select the **Data Type** of the column: String, Integer or Float
6. Select the **Index** checkbox if you will be using BMQL statements.

> By selecting an index, performance will be improved when you are using the column in a "where" clause.

7. Click **Add Column** to save the column and return to the **Data Tables Admin** page.



8. Repeat **Steps 2-7** for each column in your data table.

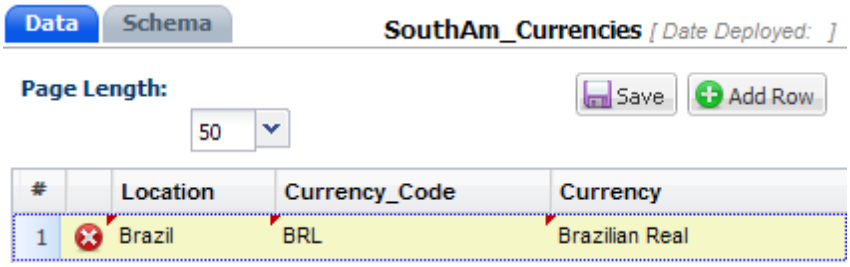
Populating the Data Table

Once you have set the schema for your table, you can begin to add the data.

1. Click on the **Data** tab in the right panel of the **Data Table Admin** page.

> The columns you created/imported in the **Schema** tab now appear on the data table.

2. Click the **Add Row** button.
3. Enter values for each column. *You can either click into each field or you can use Tab to move quickly through the fields.*

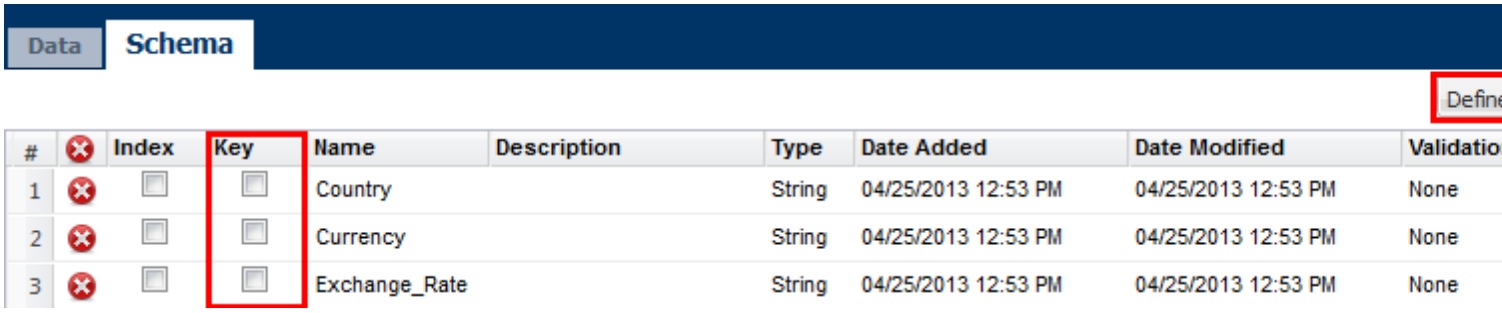


4. Repeat **Steps 2-3** until your table is complete.
5. Click **Save** once you have populated your table.

Adding Natural Keys (Delta Data Table Uploads)

BigMachines has included Natural Keys within data tables to improve efficiency. As a result of Natural Keys, the bulk upload files that modify data table data now only require the table records that are to be changed. Natural Key can be a single column or a combination of several columns that produce a unique identifier for each record (row). The creation of these Natural Keys reduces the maintenance downtime of data tables by including only data table changes upon upload. Users no longer need to purge and reload all data table rows to edit a subset of rows. For more information, please see Delta Data Table Uploads (Natural Keys) article.

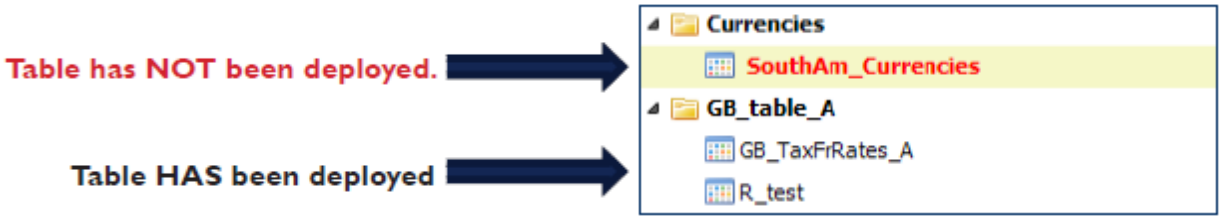
1. Navigate to the **Schema** tab.
2. Locate the **Natural Key** column.



3. Check the **Key** boxes to identify which Natural Keys will be unique identifiers OR
4. Use the **Define Keys button** to select your Natural Keys.

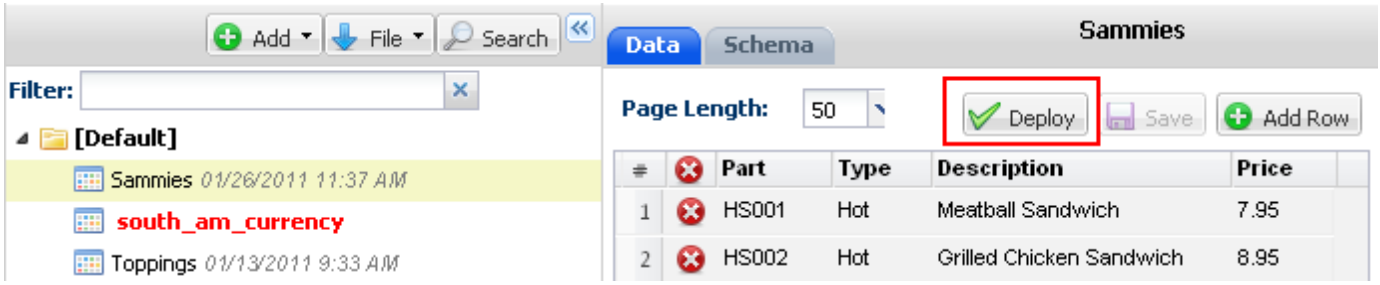
Deploying a Data Table

The final step to manually adding a table is to make sure it is deployed. All tables that have been added or modified and are NOT deployed will be clearly marked in red.

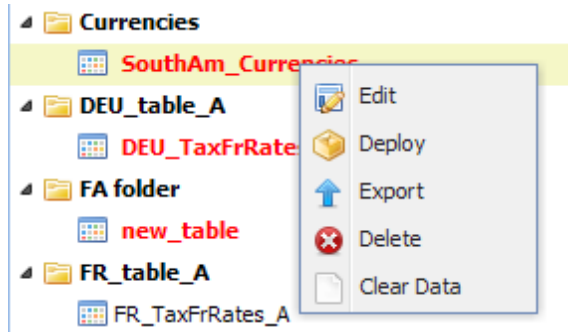


Do one of the following:

1. Click the **Deploy** button when you have a data table open.



2. Right-click on data table name. This will display multiple options for the table:



Edit: This allows you to edit the table description and view the Date Added, Date Modified and Date Deployed time stamps.

Deploy: Deploying saves the table and makes it available for use throughout BigMachines.

Export: Exports the data table to a .csv file.

Delete: Deletes the table, including its data and schema.

Clear Data: Deletes ALL table data but will keep the table and schema intact.

NOTES

To quickly populate tables, add multiple rows BEFORE entering data values.

RELATED TOPICS

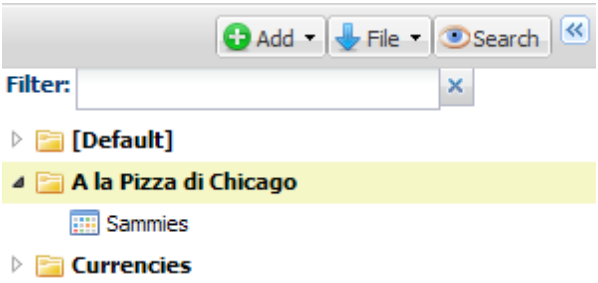
FOLDERS

OVERVIEW

Folders are used to organize data tables. All folders are located in the Folder List in the left-panel on the Data Tables Admin page. When you select a folder, it will be highlighted for you. In order to open the folder and display its data tables, double-click on the folder name.

Using Filters for Folders and Data Tables

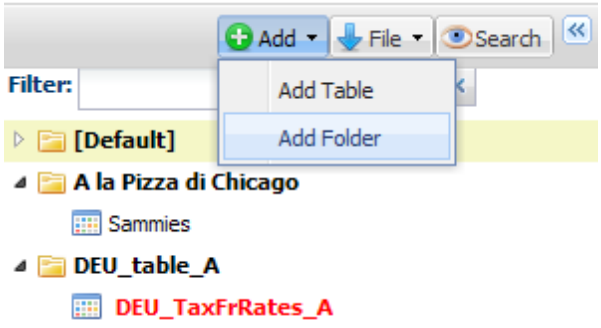
You can filter your data tables and folders by using the Filter text box. Type in a letter that could be contained in a data table name and the filter will automatically hide all folders that do not contain a data table with your filter value. For example, if you know a data table has the value "sam" in its name, you can type "sam" into the Filter and only the folders and data tables with names that contain "sam" will be visible. Select the X icon to clear results.



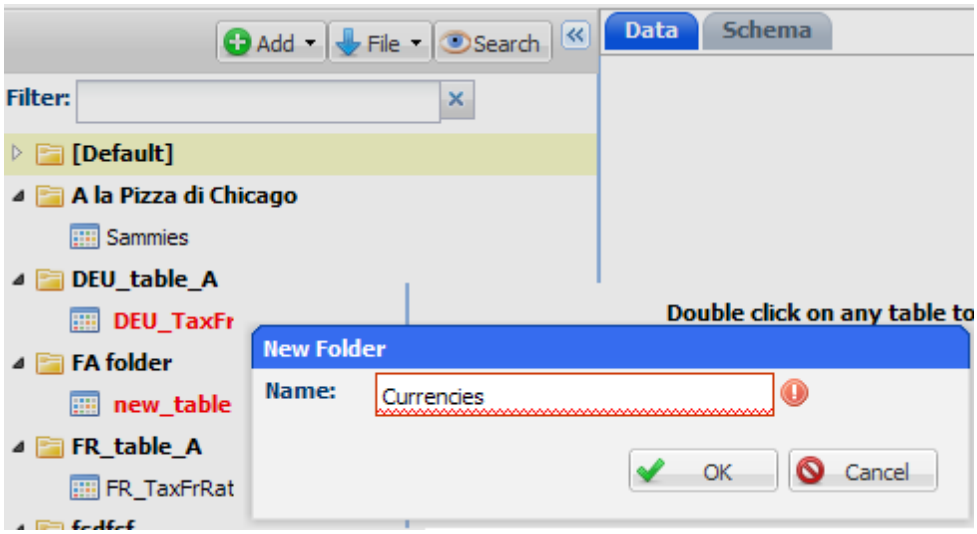
ADMINISTRATION

Adding Folders

1. From the Admin Toolbar, select **Add** and choose **Add Folder** from the drop-down menu.



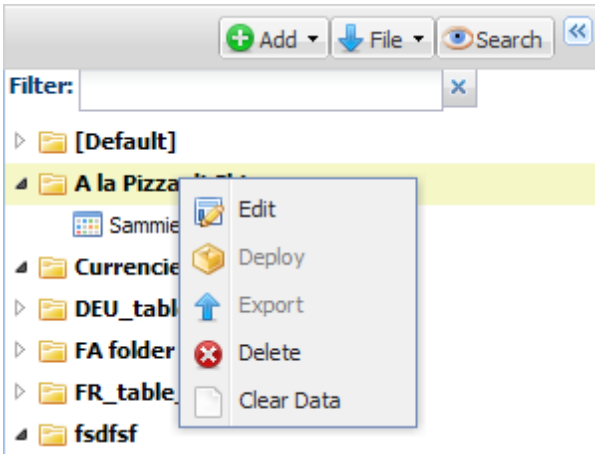
2. Enter the name for your **New Folder** and select **OK**.



The new folder will now be displayed in the Folder List.

Editing Folders

1. Right-click on a folder, a menu of editing options will appear:



RELATED TOPICS

IMPORT & EXPORT OF DATA TABLES

OVERVIEW

It is possible to maintain data tables outside of BigMachines, such as Excel (.CSV) and other file formats. We provide the functionality to allow you to import data tables and their schema directly into BigMachines. Before attempting to upload a data table, follow the instructions for setting up your .CSV file with the appropriate meta data tags for upload.

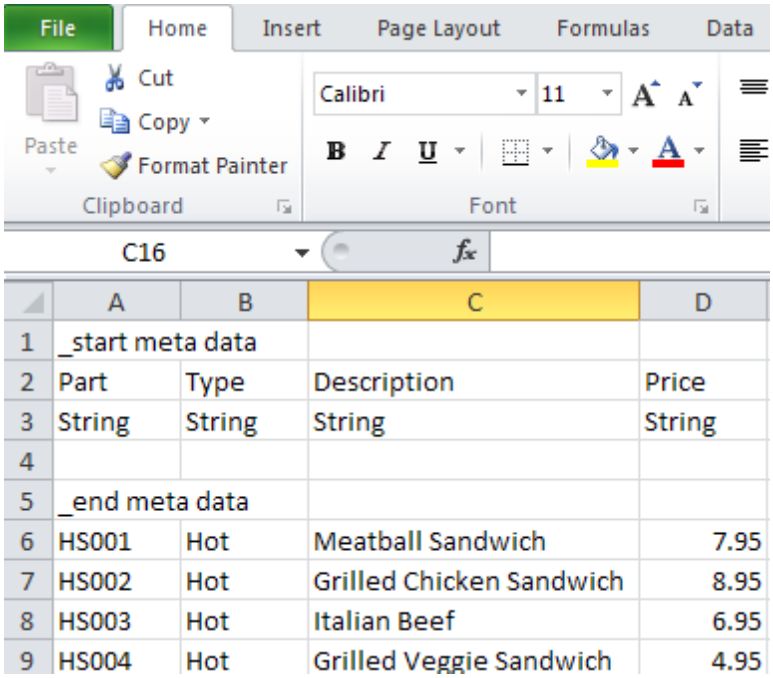
ADMINISTRATION

Preparing .CSV file for import

1. Open your Excel file or create your data table in .CSV.
2. Enter **_start meta data** into cell A1.
3. Use Row 2 for your Column Headers and Row 3 for the Data Type.

> Data types need to be either String, Integer or Float (make sure the first letter is capitalized or your import will fail).

4. Enter **_end meta data**into cell A5.
5. Populate your table. It should look like this:



	A	B	C	D
1	_start meta data			
2	Part	Type	Description	Price
3	String	String	String	String
4				
5	_end meta data			
6	HS001	Hot	Meatball Sandwich	7.95
7	HS002	Hot	Grilled Chicken Sandwich	8.95
8	HS003	Hot	Italian Beef	6.95
9	HS004	Hot	Grilled Veggie Sandwich	4.95

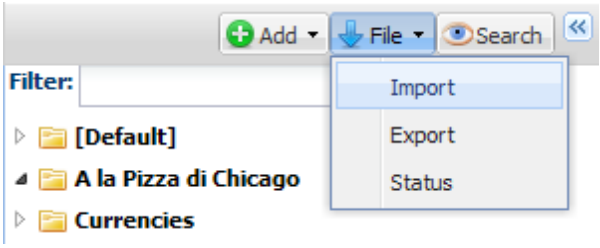
6. Save the file as a .CSV.

> Be aware that the name of your file will also be the name of your data table when it is imported into BigMachines. If a table of the same name already exists, the data will append to the existing file. Data table names can only contain alpha-numeric values and underscores. White space is not accepted.

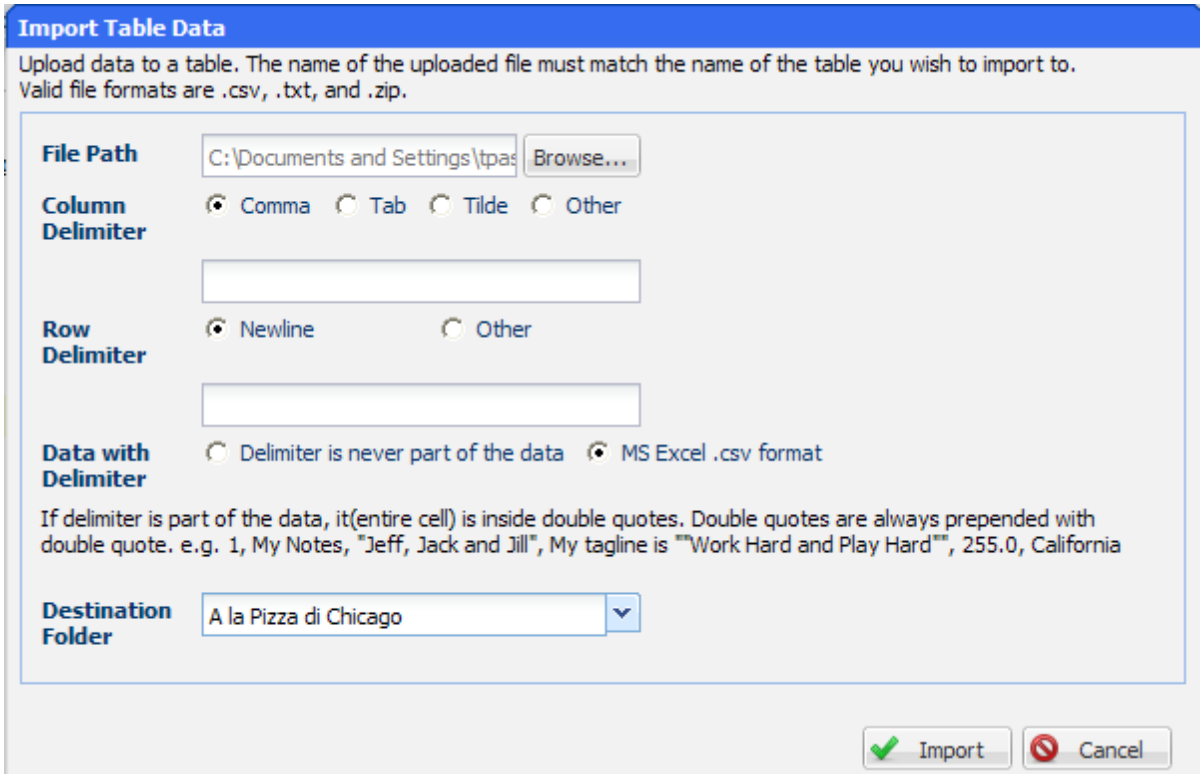
Importing a .CSV data table

Once you have properly saved and named your .CSV file, you may begin the import to BigMachines.

1. From the Admin Toolbar, click **File** and select **Import** from the drop-down menu.

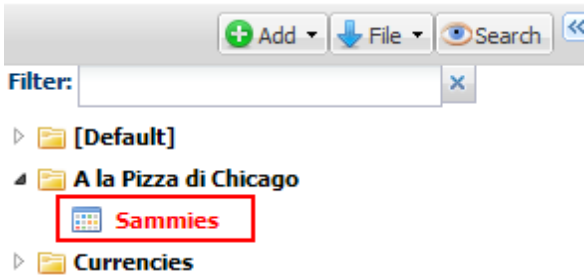


2. Click **Browse** and search for your data table.
3. Select your Column Delimiter. The default and most commonly used setting is **Comma**.
4. Select **Newline** as the Row Delimiter or enter a value and select the **Other** radio button.
5. Select **MS Excel .CSV** format option, next to **Data with Delimiter**.
6. Select the **Destination Folder** where you would like to store the data table and then click **Import**.





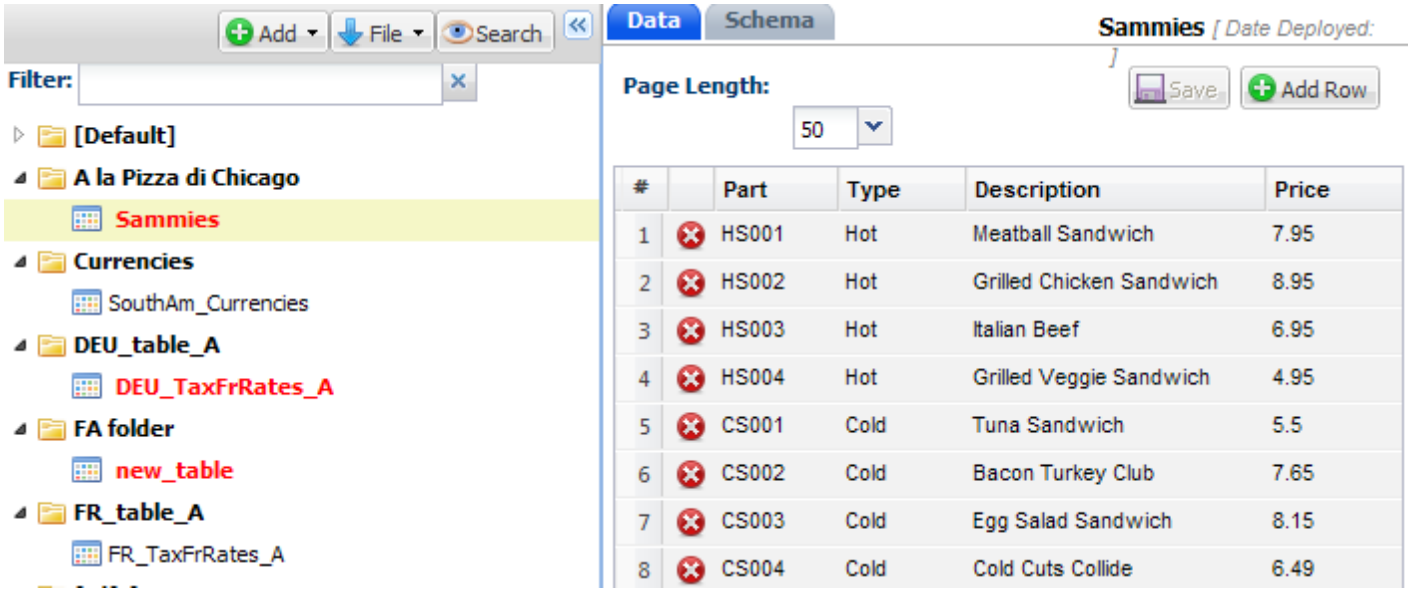
You will now see that your Data Table, based on the File Name and the Destination Folder, will be displayed in the Folder list in RED (meaning it has not been deployed).

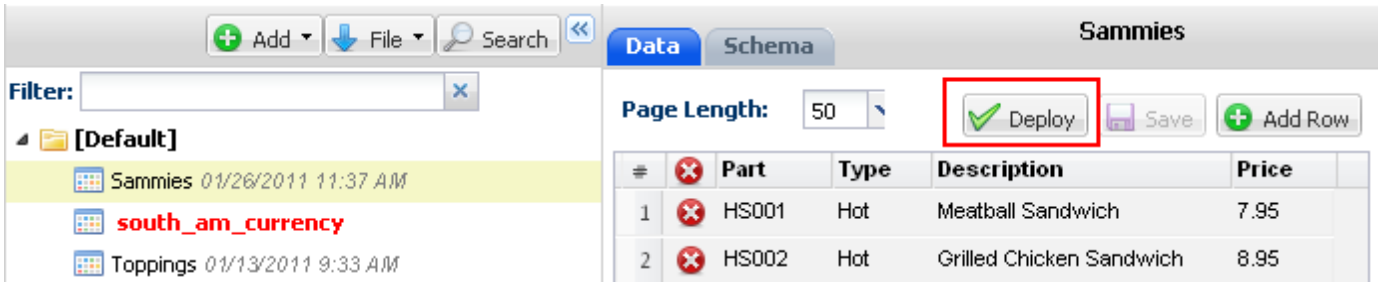


7. Deploy the table by right clicking on the data table name and clicking **Deploy**.



You can also double-click on the data table name to open the table. Here you can verify data as well as click the **Deploy** button.

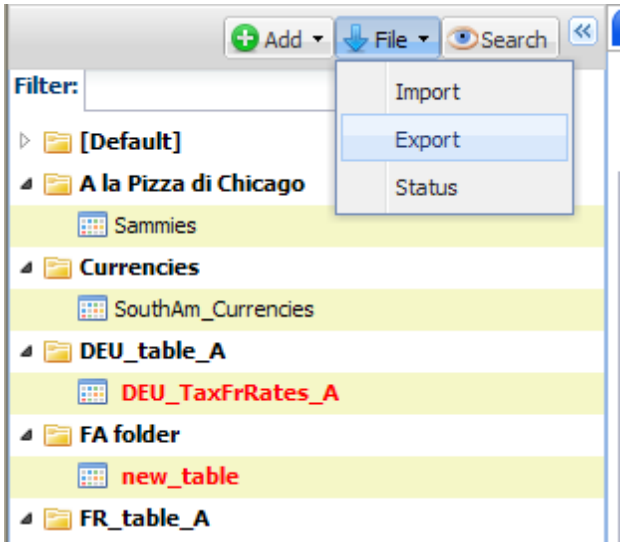




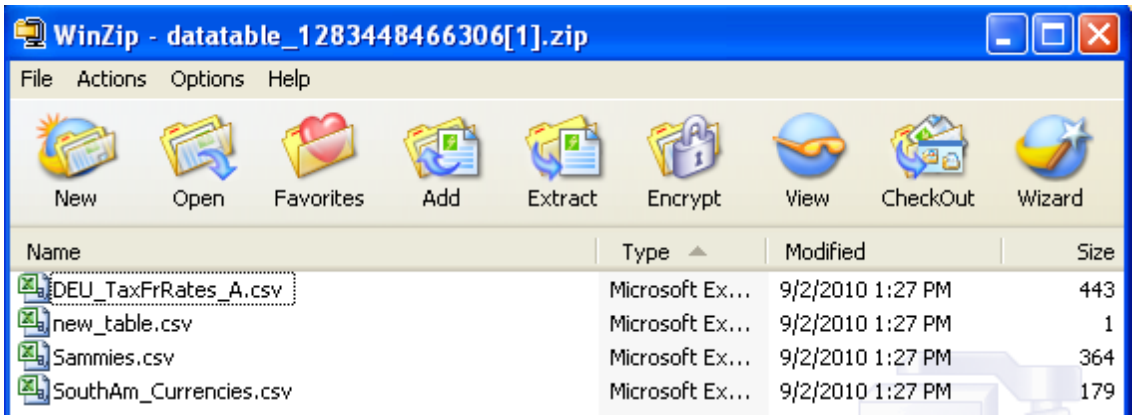
If you upload duplicate records (data with the same Natural Keys are pre-existing data in the table) during an XML/CSV upload, the duplicate records will be deleted and the new record will replace the pre-existing one. Because CSV Upload occurs row by row, if the Natural Keys match values that already exist in the table, the record will perform an update. If you try to Update a record that violates a Natural Key or doesn't exist, an error message will notify the user that the request has failed. See Natural Keys (Delta Data Table Uploads).htm for more details.

Exporting data tables

1. You can make changes to a data table and then export them back to your work station.
2. Select one table, or hold down **CTRL** button and click multiple tables, for **Export**. Selected tables will be highlighted.
3. Right-click on any highlighted data table or from the Admin Toolbar, click **File** and select **Export** from the drop-down menu.



Your file or files will be zipped and sent to your desktop or laptop.

4. 

RELATED TOPICS

DATA TABLE CHANGE LOG

OVERVIEW

BigMachines allows you to verify changes that have been made to the data tables through Bulk Download. This functionality will allow you download change logs from a particular data table, within a particular date and range, and according to some criteria (i.e. column x = XHDF). The change log files are downloaded as .CSV files. You may store up to 90 days of data table history.



This is a paid feature. Please contact your BigMachines Account Executive for pricing.

ADMINISTRATION

Example Use Case: Special Pricing on Certain Components

You work for ABC Widget Company. You have negotiated special pricing on certain components from XYZ Corporation, however these price changes are introduced periodically. You must be able to prove to XYZ Corporation that the pricing was updated at the appropriate date and time.

1. Navigate to: **Admin Homepage > Utilities > Bulk Download**.
2. Select **Data Table Audit** from the **Download Category List** and then click **Next**.

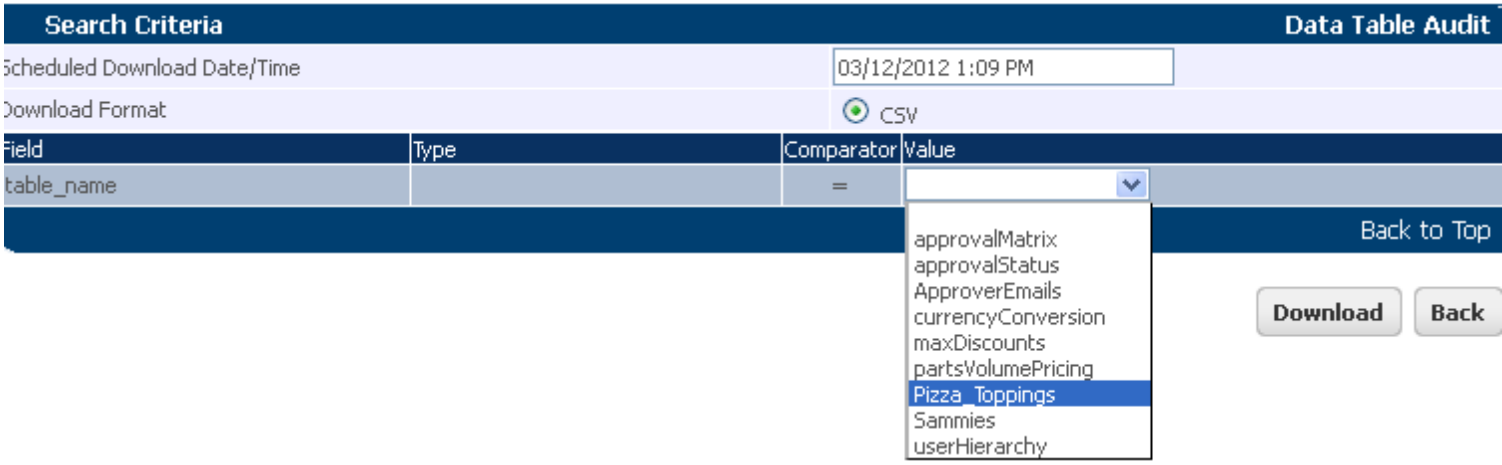
Download Category List	
Select	Category Name
○	Accounts
○	Account / User Associations
○	Commerce Transactions
○	Configurable Attributes
○	Configuration Rules
○	Data Tables
○	File Manager
○	Parts
○	Part Associations
○	Part Extended Descriptions
○	Price Book Associations
○	Serial Numbers
○	Serial Number Associations
○	User
○	Company
○	Text Administration
○	Commerce Processes
○	Library Functions
○	Product Definition
○	Printer Friendly Document
○	Translations
◉	Data Table Audit

3. Schedule a date and time to download the **Data Table Audit**.

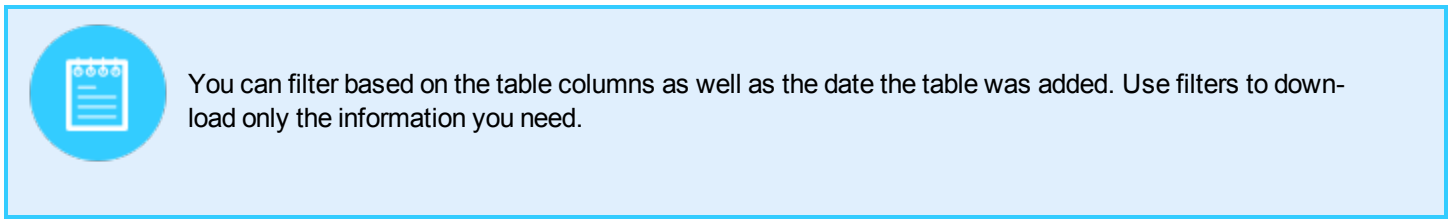


This file is only available for download as a.csv file.

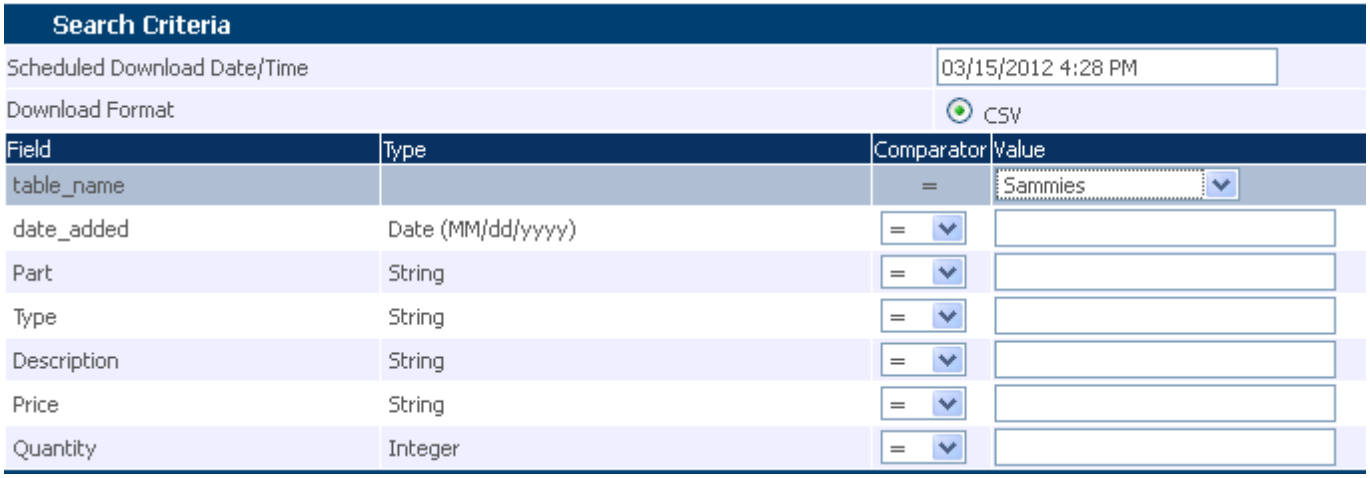
4. Select the name of the data table, for which you must download the **Data Table Audit** file, from the drop-down menu.



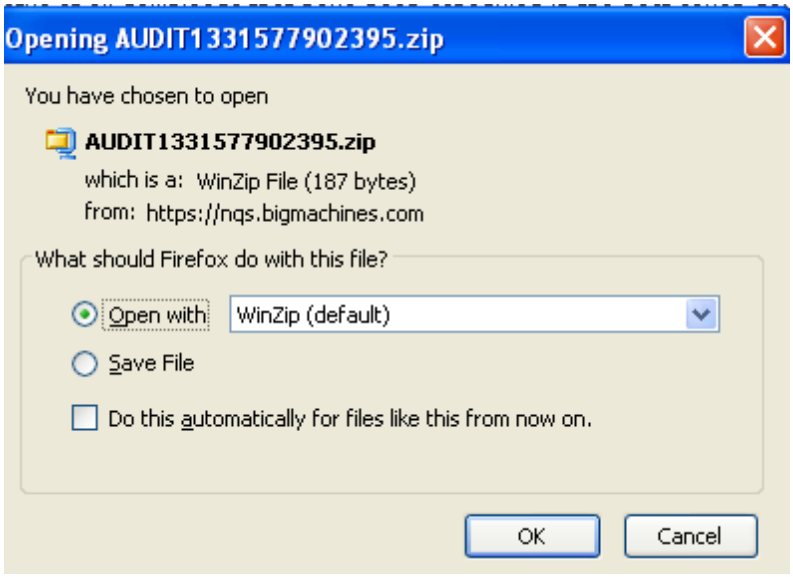
5. Select to filter the change log data, if desired.



You can filter based on the table columns as well as the date the table was added. Use filters to download only the information you need.



6. Click **Download**.
7. Double-click on the **ScriptTableAudit** module from the **Download Status** page.
8. Open the zip file.



9. Open the file and review the change log.

> The file will only show you table rows that have been changed and who changed it, but will not display previous values.
>
> An audit record is created for every change. If you filter a particular column value, you'll see all of the manipulations on that row. The only issue that this might create is if data has not been updated within the last 90 days.

6	HS001	Hot	Meatball Sandwic	7.95	1	INSERT	superuser
7	HS002	Hot	Grilled Chicken Sa	8.95	1	INSERT	superuser
8	HS003	Hot	Italian Beef	6.95	1	INSERT	superuser
9	HS004	Hot	Grilled Veggie Sar	4.95	1	INSERT	superuser
10	CS001	Cold	Tuna Sandwich	5.5	1	INSERT	superuser
11	CS002	Cold	Bacon Turkey Clut	7.65	1	INSERT	superuser
12	CS003	Cold	Egg Sald Sandwich	8.45	1	INSERT	superuser
13	CS004	Cold	Cold Cuts Coldie	6.5	1	INSERT	superuser
14	CS004	Cold	Cold Cuts Coldie	6.5	1	DELETE	superuser
15	WS001	Warm	Bacon Turkey Brav	5.99	1	INSERT	superuser
16	WS002	Warm	Philly Steak and C	5.99	1	INSERT	superuser

NOTES



Each modification to a data table record will result in an audit record or a row in the change log. This means each individual row that is inserted, modified, deleted, or deployed will result in a row in the change log.

If you are using data table import and export (or bulk upload/download), the logs will appear as if all data is deleted and re-added. For example: if you have a data table with 100 rows, when you delete all records and bulk upload a new table, you will see 200 rows in the change logs.



You cannot clear a data table change log. Data older than 90 days will not be found in the data change log.

Log a case with BigMachines Support Portal to enable the data table audit functionality. **Turning on this feature may negatively affect the performance of bulk data table uploads.**

Column data type changes are not recorded in the data audit log file. So, if you make changes to the data type of a data table column, these changes will not be reflected in the change log.



Do NOT turn this feature on unless it is necessary (due to reduced data table performance).

RELATED TOPICS

DATA TABLE VALIDATION

OVERVIEW

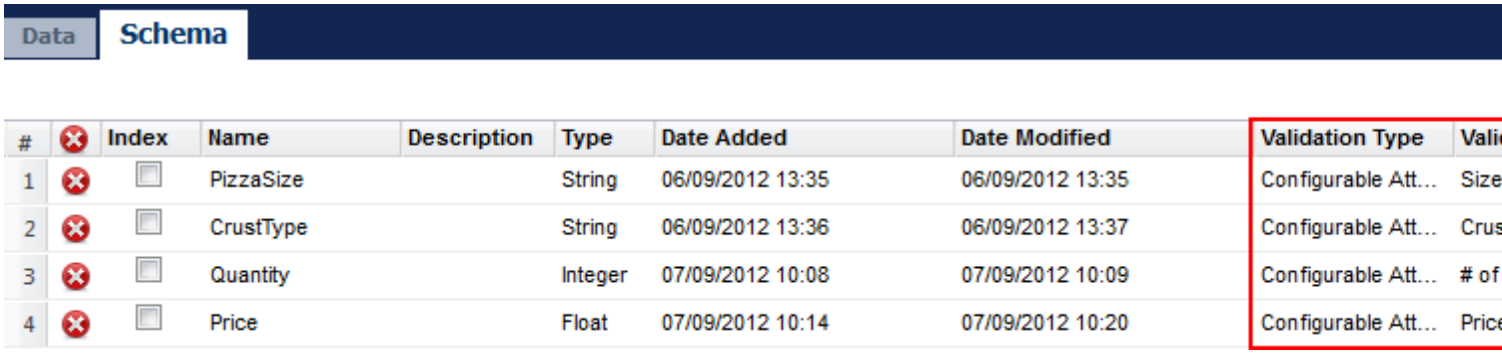
BigMachines enables users to validate table data by mapping commerce and/or configuration attributes to a data table column. When you add this functionality, the validation will determine if the table data is valid or invalid. Column data must match one menu value in the linked attribute.

ADMINISTRATION

Setting Up Data Table Validation

1. Navigate to: **Admin Home Page > Data Tables > Select a Table**.
2. Click the **Schema** tab.
3. Verify the column information is correct (if legacy) or add columns.
4. Select a **Validation Type** and **Validation Mapping** for each column.
    1. **Validation Type:**
        1. **None:** Select if you do not wish to perform a validation on this column.
        2. **Commerce Attribute:** Select if mapping to a commerce menu attribute.
        3. **Configurable Attribute:** Select if mapping to a configurable menu attribute.
    2. **Validation Mapping**
        1. If validation type is 'None', this field will be unavailable for mapping.
        2. If validation type is 'Commerce Attribute', select from main-document or sub-document menu attributes.

           If validation type is 'Configurable Attribute', select from configurable menu attributes in the product hierarchy.





Integer and float multi-select menus can be used in a validation, but only a single value can be selected. Saving schema might take a long time if a column is mapped, or will be mapped, to a multi-select menu attribute.

5. Click **Save**.



For legacy data, you can add and save the validation, but you will get an error message that instructs you to remove the invalid data before you can deploy the data table.

TROUBLESHOOTING

When adding a validation to a data table with existing data, and invalid data is present, a button named 'Invalid Data' will allow you to:

- **Delete all invalid data.** When data is deleted, it will delete the entire row.
- **Display which records are invalid OR export to a CSV.** Using .CSV, you can update the data and import the correct .CSV file.



Import .CSV will not update existing records. It will allow you to modify values. To replace the table, you must clear the data table and upload the file. Alternatively, you could delete the table and upload a new one through .CSV.

NOTES



Validations will run as part of bulk import. If there are records that do not meet the validation, then the invalid records will fail.

We do not validate column data type with Parent Entity data type because in Commerce, we only have text type menu attributes.

To successfully migrate a data table that has one commerce menu attribute validating one column and one configurable attribute validating another column:

1. Performance is acceptable, but there will be a decrease in performance on both the import and the UI if large data tables are being uploaded.
2. The indication in the logs will be the record number in the .CSV file.
3. Remove either the commerce or configuration validation, migrate the data table, and add the validation back to the migrated table.

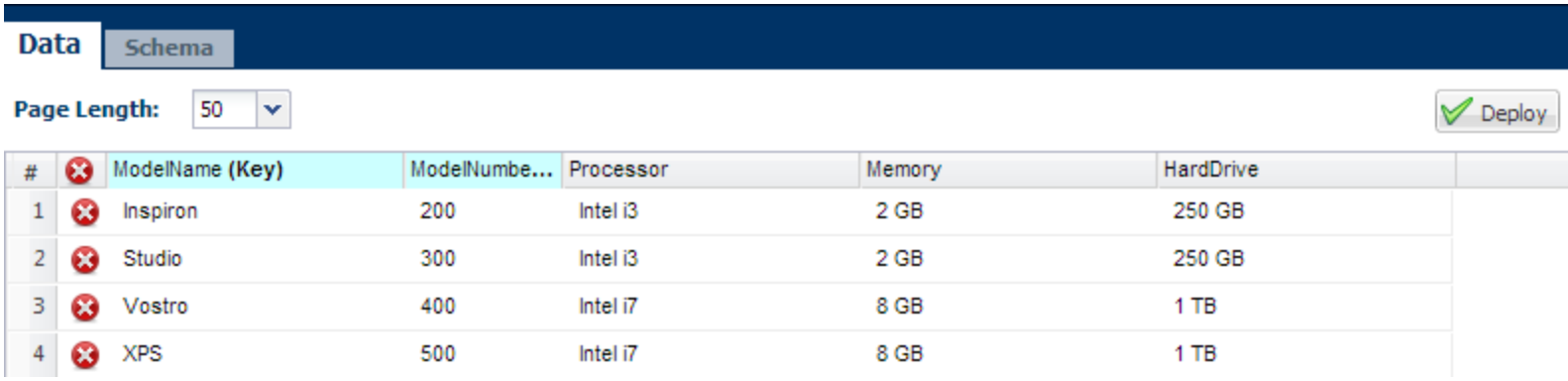
RELATED TOPICS

NATURAL KEYS (DELTA DATA TABLE UPLOADS)

OVERVIEW

BigMachines has included Natural Keys within data tables to improve efficiency. As a result, the bulk upload files that modify data table data now only require the table records that are to be changed. Natural Keys can be a single column or a combination of several columns that produce a unique identifier for each record (row). Natural Keys create a unique set of data (contained within rows) by selecting specific values (contained within columns) to identify that data. The ability to do delta data table uploads, which allow only altered data table rows to be updated (rather than require a user to delete and re-upload an entire table to make a small change), reduces the maintenance downtime of data tables. Users no longer need to purge and reload all data table rows to edit a subset of rows.

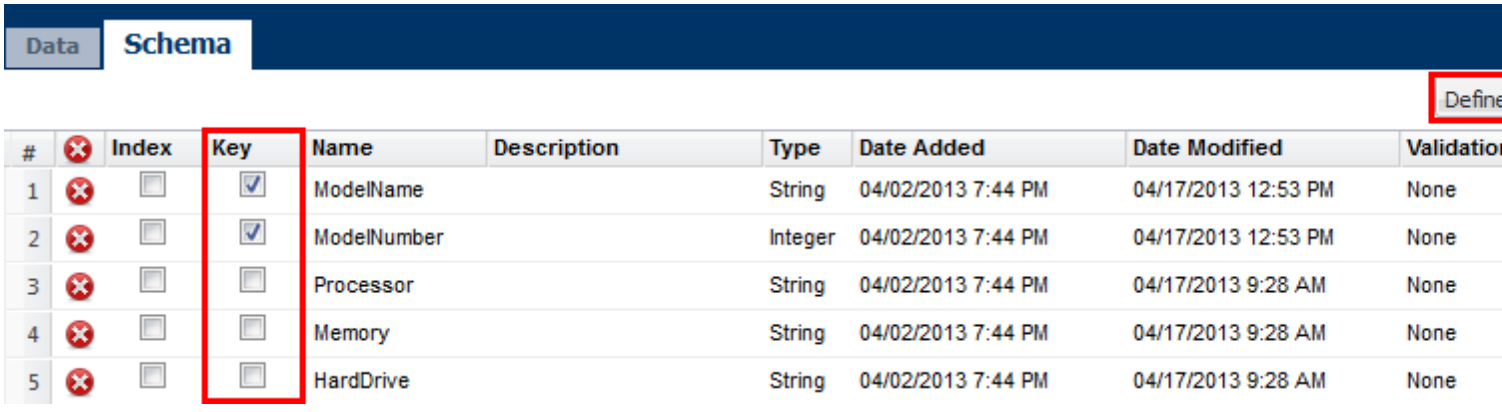
Natural Keys are supported in: Data Tables, Web Services API, CSV Upload and Bulk Upload.



ADMINISTRATION

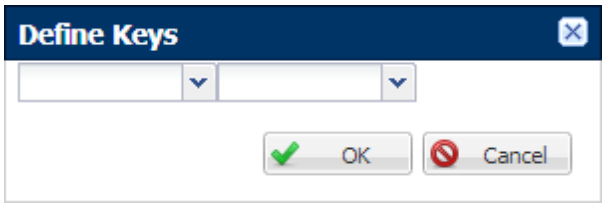
Adding a Natural Key to a Data Table

1. Select the desired **data table.**
2. Navigate to the **Schema** tab.
3. Locate the **Natural Key** column.



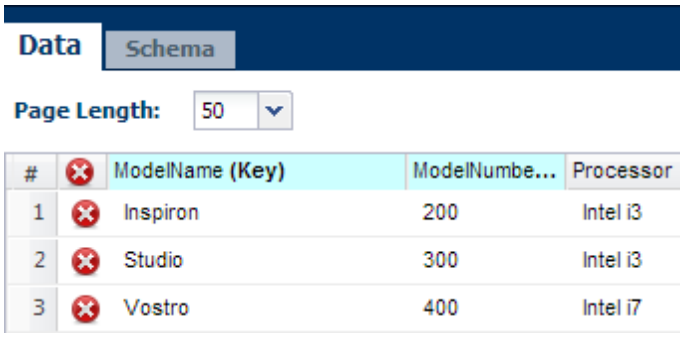
4. Check the **Key** boxes to identify which Natural Keys will be unique identifiers OR

5. Use the **Define Keys** button to select your Natural Keys.



If using the Define Keys button, the following pop-up window will appear where user can select desired natural keys:

6. Ensure that the added natural keys are valid. Columns with valid Natural Keys will look like this:



TROUBLESHOOTING

Invalid Data

If a Natural Key is not unique, for example two or more rows contain identical value(s) in the column(s) designated as Natural Keys, the record is considered invalid. Invalid cell(s) will be highlighted in red.

There are two scenarios where invalid data cells would be highlighted in red:

20 Latitude 100	Existing data in a saved table becomes invalid upon applying natural keys. The cell is highlighted in red, while the text remains black.
22 Latitude 100	Duplicate data is introduced to a Key column. The cell is highlighted in red and the text color is also changed to red.

If invalid data is identified after a Natural Key is applied within existing data, you can still save the table. However, you will be unable to deploy the data table until all invalid data is reconciled.

If in valid data is identified when the user is setting up a new data table, or adding new data to an existing table with a Natural Key, you will be unable to save the table.

Invalid data & uploads

If you attempt to upload duplicate records (data with the same Natural Keys are pre-existing data in the table) during an XML/CSV upload, the duplicate records will be deleted and the new record will replace the pre-existing one. Because CSV Upload occurs row by row, if the Natural Keys match values that already exist in the table, the record will perform an update. If you attempt to Update a record that violates a Natural Key or doesn't exist, an error message will notify the user that the request has failed.

NOTES

When using the Data Table Natural Keys feature, it is strongly recommended that no more than five string columns be assigned to a part of the key.

The maximum setting is 10 Natural Keys in a single data table. Please contact BigMachines if you require additional Natural Keys.

If more than six string columns are assigned to the Natural Key, an error message appears ("The call failed on the server; see server log for details") and the columns marked as keys will not display correctly. *To fix this issue, admin must remove the Natural Keys from all columns and Save the Data Table. Once this is done, Natural Keys can be added to six or fewer columns.*

Legacy Customers: When modifying a data table via XML Upload be mindful of row ID tags. Since an ID is already associated with your row, uploading an XML version with an ID may identify the row twice. If you choose not to use Natural Keys, you must use the row ID when modifying your table.

RELATED TOPICS

UPLOAD STATUS & ERROR LOGS

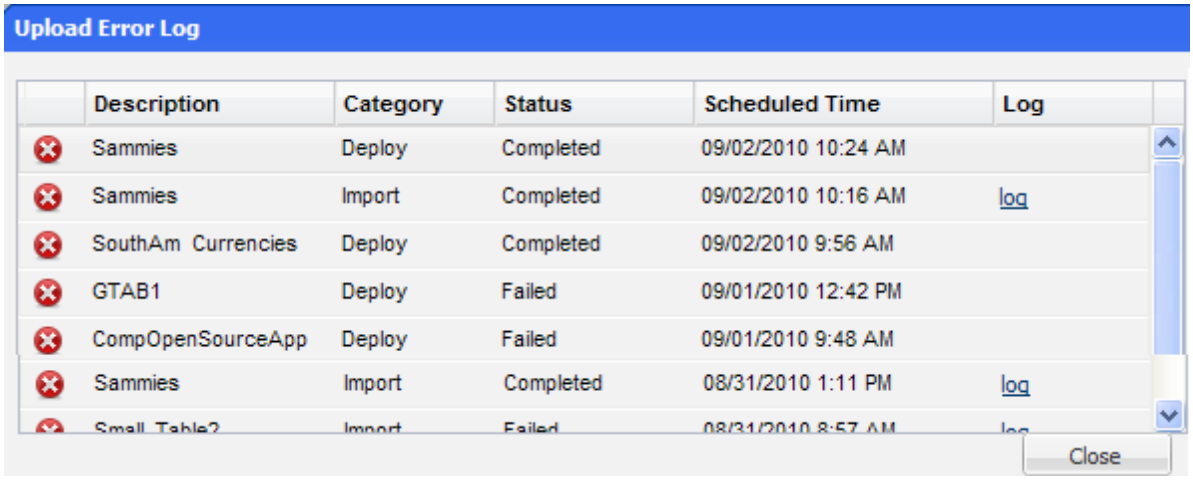
OVERVIEW

After you have imported or deployed a data table, you can check the status and corresponding error logs to ensure that all of your data loaded correctly.

ADMINISTRATION

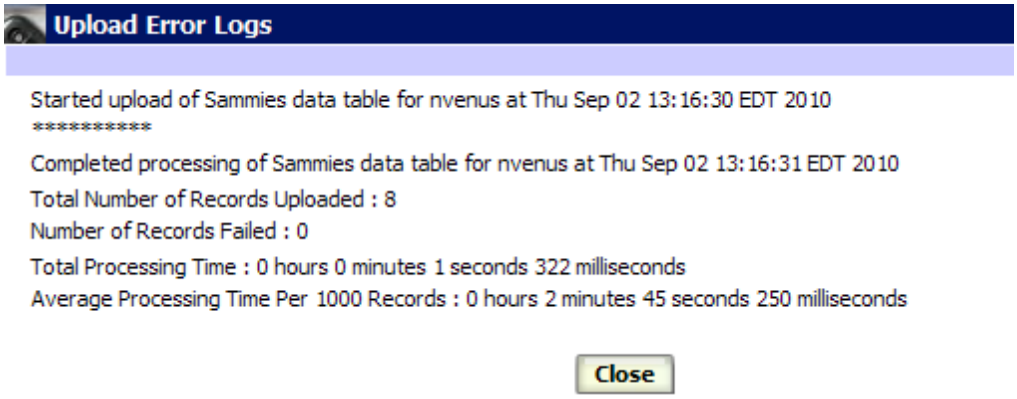
Viewing Error Logs

1. Click **File** from the **Admin Toolbar** and choose **Status** from the drop-down menu.



2. Check the status of each data table import or deployment.

3. If available, click the **Log** link to check the error logs.



NOTES



CSV Upload occurs row by row, so if the Natural Keys match values that already exist in the table, the record will perform an update.



If you attempt to upload duplicate records (data with the same Natural Keys already exist in the table) during an XML/CSV upload, the duplicate records will be deleted and the new record will replace the pre-existing record.

If you update a record via Web Services, that violates a Natural Key or doesn't exist, an error message will notify you that the request has failed.

RELATED TOPICS

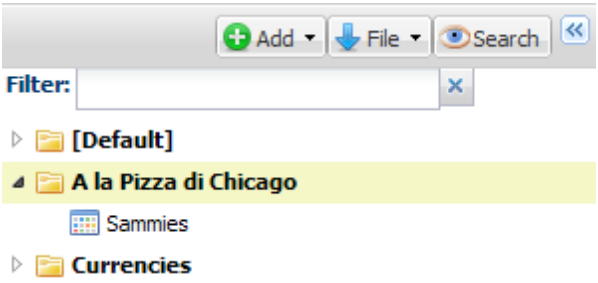
FOLDERS

OVERVIEW

Folders are used to organize data tables. All folders are located in the Folder List in the left-panel on the Data Tables Admin page. When you select a folder, it will be highlighted for you. In order to open the folder and display its data tables, double-click on the folder name.

Using Filters for Folders and Data Tables

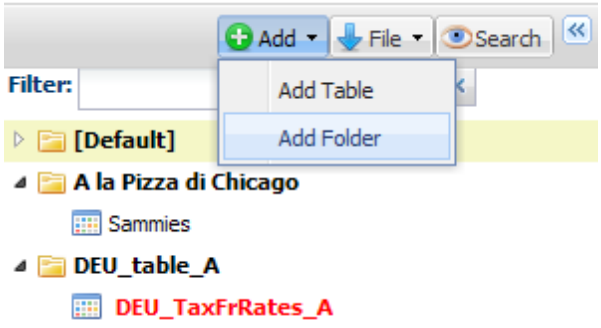
You can filter your data tables and folders by using the Filter text box. Type in a letter that could be contained in a data table name and the filter will automatically hide all folders that do not contain a data table with your filter value. For example, if you know a data table has the value "sam" in its name, you can type "sam" into the Filter and only the folders and data tables with names that contain "sam" will be visible. Select the X icon to clear results.



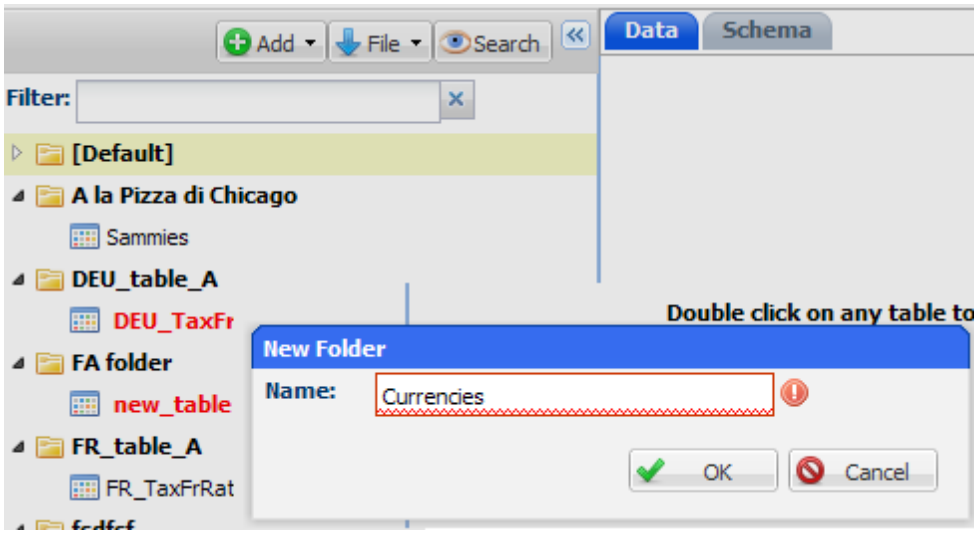
ADMINISTRATION

Adding Folders

1. From the Admin Toolbar, select **Add** and choose **Add Folder** from the drop-down menu.



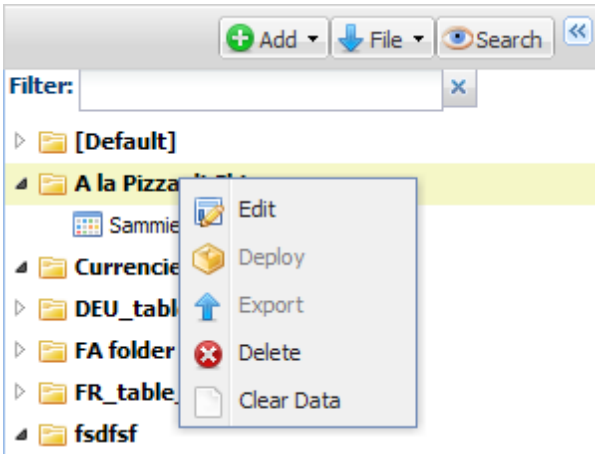
2. Enter the name for your **New Folder** and select **OK**.



The new folder will now be displayed in the Folder List.

Editing Folders

1. Right-click on a folder, a menu of editing options will appear:



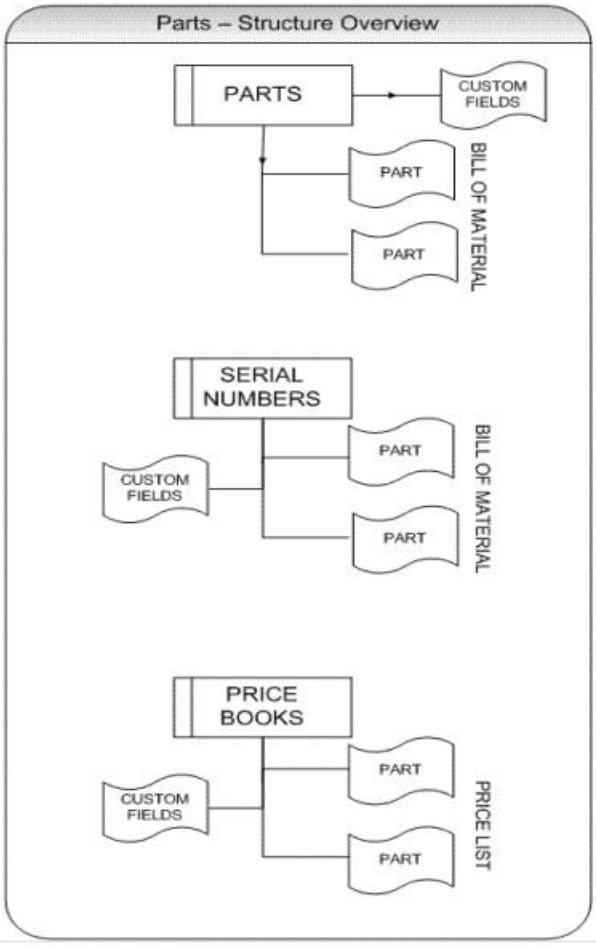
RELATED TOPICS

PARTS - AN OVERVIEW

OVERVIEW

Parts data includes part numbers, serial numbers, part associations and serial number associations. There are two ways you can create parts data: through a bulk upload process or through manual entry.

Parts Structure



Bulk Upload

You can enter parts data, price book and serial number data through upload. The formats supported for upload are XML and CSV. This process enables you to enter large batches of parts data into the system.

Manual Entry

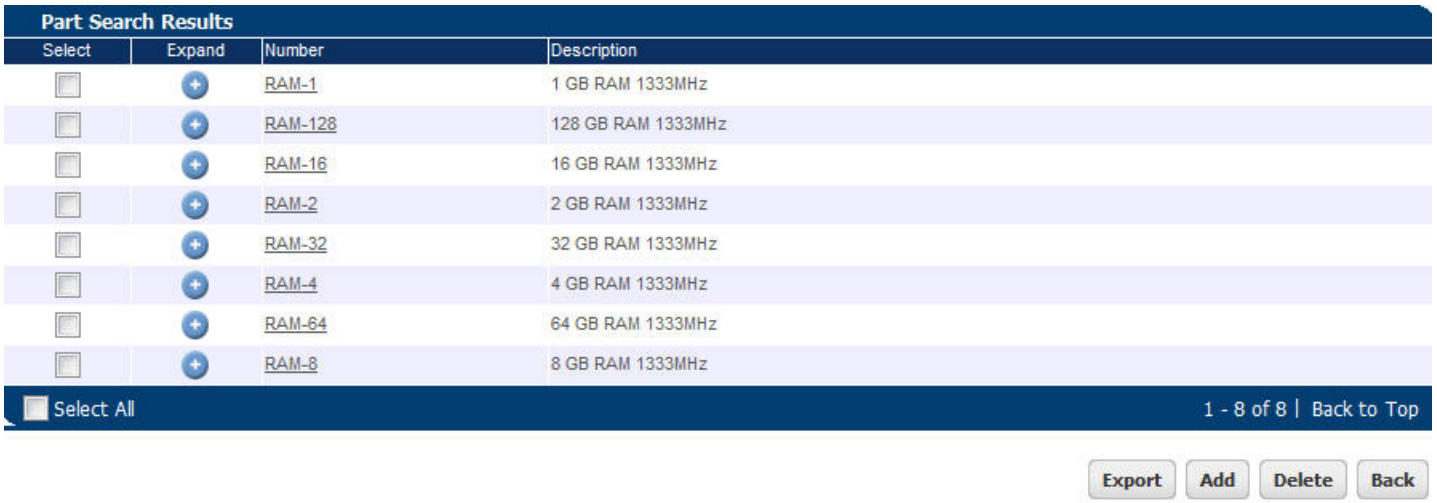
Parts data can also be administrated manually. Within Parts Administration, you can also set the visibility level of BOMs (Bills of Materials) to regulate how much parts data buyers see. We recommend using the bulk upload process for entering large amounts of parts data. For small amounts of data, manual entry in the parts administration area may be a quicker process.

ADMINISTRATION

Searching for Part Numbers

You can search for parts by using serial numbers or parts.

1. Navigate from Admin Home Page > Products > Parts
2. Enter a value into the Serial Number or Part Number text box.
3. Click **Search**.
4. Click the expand icon to see any Part or Serial Number associations.



5. Click **Export** to download a .zip file and Excel .csv file containing the parts list.

If you only know a partial part number, you can search by the first few characters. The search engine locates all parts starting with the characters entered. For example, you can conduct a search using "123", and the search engine will find parts such as "12330", "12357", "123a1"... etc.

Setting BOM Levels

The BOM level restricts how many levels of a BOM users can see. Set this option if you want to configure how many levels to show. By default, users can select up to 15 levels of BOM, the maximum number of levels.



NOTES



Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.*

RELATED TOPICS

PARTS AND ASSOCIATIONS

OVERVIEW

Parts data can be managed manually. We recommend using the bulk upload process for entering large amounts of parts data. For small amounts of data, manual entry in the parts administration area may be a quicker process.

ADMINISTRATION

Adding a Part

1. Navigate from Admin Home Page > Parts > Add New Part.
2. Enter a unique **Part Number.**
3. Complete information in the **Product Information Section**:
    - **Direct Buy:** Choose a buy type using the drop-down menu.
        - **Direct Buy:** Part can be purchased alone
        - **Buy Parent Part:** Part must be purchased along with a parent part
        - **Buy as Recommended Item Only**: Part can only be purchased through a recommended item rule.
    - **Description:** Enter a plain text description for the part.
    - **Extended Description 1 and 2:** Enter plain text or formatted HTML part descriptions.

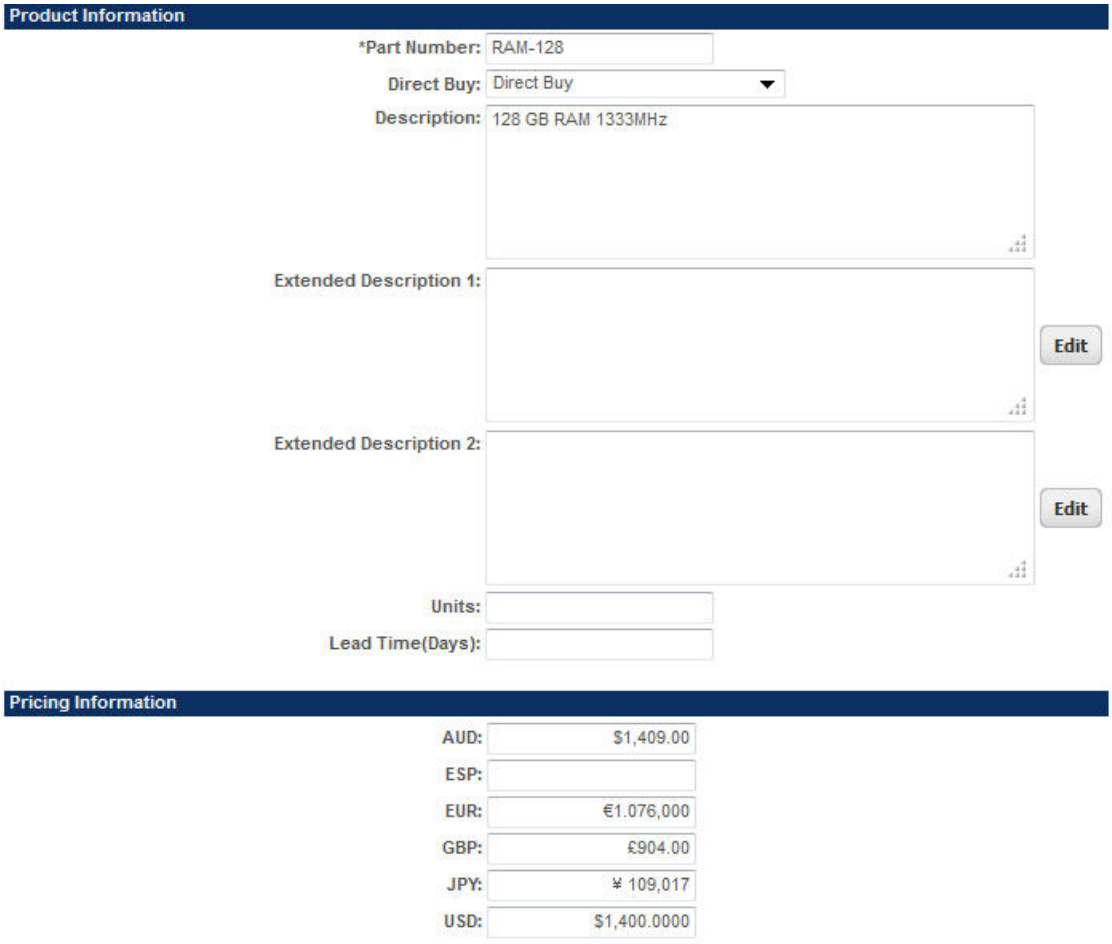
      Clicking Edit displays the HTML Editor that enables you to enter and edit a subset of HTML. You can also link images to the HTML and generate image rollovers.

    - **Units:** Enter the number of units for the part number.
    - **Lead Time:** Enter the Lead Time in days. This is the time taken for the part to be shipped.
4. Complete **Product Integration Information** by entering the Partner Part Number ID and standard Price Book, if necessary.
5. Enter a **part price** for each of your supported currencies.



When a price exists in your base currency, but no other currencies, pricing will be available in those other currencies provided that exchange rates exist.

6. Enter values for the **custom parts fields** in the Extended Information section. See Customizing Part Fields below for more information.
7. Click **Browse** to upload and associate up to 4 square images with the part for File 1, File 2, File 3, and File 4 fields.
8. Click **Add**. This adds the new part with the details provided, and displays the Part Search Results page that lists the part numbers matching your search.



To edit a part, search for the Part on the Parts Administration page and click the Part Number.

Deleting a Part Number

1. Navigate from Admin Home Page > Parts > Search for Part by Part Number.
2. Select the checkbox corresponding to the serial number to be deleted.

3. Click Delete. The refreshed view of the Part Number Search Results page shows the deletion.

> Use caution when deleting parts as they cannot be recovered. All parts associations are deleted as well.

Associating Parts

Parts can be associated with one another in a parent-child relationship.

1. Navigate from Admin Home Page > Parts > Search for Part by Part Number.
2. Search for the part whose association you want to create, or create a new part.
3. Click the expand icon . This will take you to the Part Tree for Part Number page.



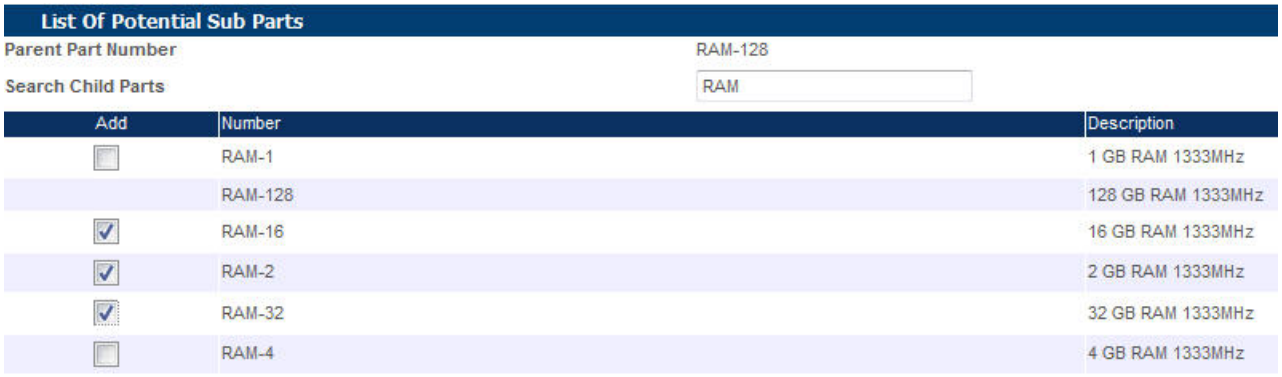
Part Search Results

Select	Expand	Number	Description
☐	⊕	RAM-1	1 GB RAM 1333MHz
☐	⊕	RAM-128	128 GB RAM 1333MHz
☐	⊕	RAM-16	16 GB RAM 1333MHz
☐	⊕	RAM-2	2 GB RAM 1333MHz
☐	⊕	RAM-32	32 GB RAM 1333MHz

4. Click **Add Subnode**.
5. Search for the parts you want to associate.

> Part associations can only be added as first level sub-parts.

6. Create associations by marking the checkboxes corresponding to the parts, and then click **Associate**.



To delete an association, navigate back to the Part Tree for Part Number page, select the parts associations you want to remove, and click delete. It is not possible to undo a parts association deletion.

Customizing Part Fields

Custom part fields can be used to create part filters and/or to supplement the set of base attributes available for parts. BigMachines supports up to 30 custom part fields. These fields are pre-defined and can be hidden or displayed on the search pages.

Fields

Of the 30 fields:

- **Text:** There are 25 text fields that can be of String or Single Select Menu data types.
- **Text:** There are two text fields that can be of multi-select Menu data types.
- **Number:** There are three number fields that can be of Float or Integer data types.

---

1. Standard Navigation Path: Admin Home Page > Parts > Customize Part Fields
2. Choose either a **text, multi-select menu or number field** and then choose the field type using the drop-down list, if provided.
3. Click on the **Create Field** button.
4. Enter the values for the available fields based on data and field type:

Element	Data Type	Field Type	Description
Field Name	Number and Text	All available	Enter a name for the custom part field. This is a required field.
Default	Number	Float and Integer	Enter a default value that will pre-populate the custom part field. This is a required field.

Element	Data Type	Field Type	Description
Value			
Required	Number and Text	All available	Select if the fullaccessuser must enter a value into the custom part field when creating new part numbers.
Show Field	Number and Text	All available	Select user-side pages that will display the custom part field. Options include Search Page, Search Results Page, BOM page, and Detail Page.
Precision	Number	Float	Choose the number of decimal places that will be accepted into the custom part field.
Validation Fields	Number	Float and Integer	Choose validation type of either: Check Float in Range or Check for Positive Float. You can also enter a minimum and maximum value.
Unit Fields	Number	Float	Select a conversion group and its associated default unit for the custom part field. .
Search Operator	Number	Float and Integer	Select an operator for the custom part field.
Search Behavior	Text	String	Choose both a leading and trailing wild-card. Example: 'thisisapart', and you're searching for 'isa': • Implicit: Will automatically result. • Explicit: Must type wild-card to see result, ie. %isa%.
Menu Population	Text	Single Select Menu and multi-select Menu	Populate menu values. Click here for more information.

5. Click **Add**. The part will now appear in the Part Define Fields Section.
6. Choose a **custom part field** and set the **field type** to **Filter**in order to create part filters. Otherwise leave as **Normal**.
7. Click the **Deploy** button to activate the custom parts fields and filters.

NOTES



Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.*



Part Associations can only be added as first sub-parts.

Part Associations are only used when viewing a Part within Configuration.



This parent-child relationship does not recommend or add Parts.



It is not possible to undo a Part Association deletion.

The association will not be viewable on a quote within Commerce because this is strictly Configuration functionality.

RELATED TOPICS

CREATING FILTERS

OVERVIEW

Custom part fields can be used to create part filters and/or to supplement the set of base attributes available for parts. BigMachines supports up to 30 custom part fields. These fields are pre-defined and can be hidden or displayed on the search pages. Of the 30 fields:

- **Text:** There are 25 text fields that can be of String or Single Select Menu data types.
- **Text:** There are two text fields that can be of multi-select Menu data types.
- **Number:** There are three number fields that can be of Float or Integer data types.

Any of the fields can be used as filter attributes. Part filters allow you to control access rights to the parts. In order to create a parts filter, you will need to identify parts with certain custom values and then create permissions to each classification. **Example:** you can create a filter called "brand" . You can create filter rules so that American users can see parts that are of type Brand1 while users in Europe can only see parts that are type Brand2. These filter rules can be based on user name, company name, user type, company type, user groups and Customer ID.

ADMINISTRATION

Creating a Parts Filter

1. Navigate from: Admin Home Page > Products > Parts
2. Click **Customize Parts Fields**.
3. Create a custom part field, if necessary.
4. Choose a **Custom Part Field** and set the **Field Type** to **Filter**.
5. Click **Update** .
6. Click **Define Script** next to the custom part you chose.
7. Write a BML script to define the filter. *The following attributes are available as inputs to the filter:*

Attribute Name	Variable Name	Data Type	Description
User Login Name	**_user-name**	String	Restrict or allow access to parts by selecting a specific user login to add to your filter script. I.E. if (_username = "joe-smith") { return ""};
Company Login Name	**_company**	String	Restrict or allow access to parts by selecting a company name to add to your filter script. I.E. if (_username = "joe-smith" and _company = "BigMachines") { return ""};
User Type	**_user_ type**	String	Restrict or allow access to parts by selecting a user type to add to your filter script: RestrictedAcces, FullAccess, Quick-RegistrationBuy, ChannelAgent, SalesAgent and/or BuyAccess. I.E. if (_username = "joesmith" and _company = "BigMachines" and _user_type = "ChannelAgent") { return "" };

Attribute Name	Variable Name	Data Type	Description
Company Type	**_company_type**	String	Restrict or allow access to parts by selecting a company type to add to your script: RestrictedAccess, FullAccesWithESales, BuyAccess, and/or QuickRegistrationBuy. I.E. if (_username = "joesmith" and _company_type = "FullAccessWithESales") { return ""};
User Groups	**_user_groups**	String	Restrict or allow access to parts by selecting a user group to add to your filter script. I.E. if (_user_groups = "Girl Scouts) { return Acme Parts; } return "";
Customer ID	**_customer_id**	String	Restring or allow access to parts by selecting a Customer ID.

8. Click **Check and Save**.
9. Click **Deploy** to activate the custom parts fields and filters.

The following example script shows how parts filtering can work:

```
//Logic: Filter parts based on company's user type
//Field returned: String
if (_company_type=="BuyAccess" and _user_type=="FullAccess"){return ""};
return "Acme Parts, Generic Parts";
```



This script example could be used to filter parts based on company type and user type. FullAccess users in the BuyAccess company type see all classifications of parts because all menu values are returned with the double quotes. Any other user type sees parts classified as "Acme Parts" or "Generic Parts".

Removing Filters

You can remove a filter from a defined field in one of two ways. *To delete the filter, you can remove it from the system. To shut off the filter, without deleting it, you can toggle the field type from filter to normal.*

1. Choose the field whose filter you want to remove, and do either of the following:
   - **To shut off the filter**: Select **Normal** under the Filter/Normal column, and click **Update**.
   - **To delete the filter**: Select the **Delete** checkbox, and click **Update**.
2. Click **Deploy** to activate filter changes.

USE CASES

Inactivating parts without removing from Parts Database

You need to mark some parts as inactive and remove them from use without removing them from the parts database.

1. Add a new single-select menu field using the steps under "Creating a Parts Filter".
2. Create a require field with the options: Active and Inactive. *Set the appropriate default value for your case.*
3. Write return "active"; in the filter script. The new field will be blank for all existing parts. You must set it to the desired value for ALL EXISTING PARTS.

NOTES



Parts filtering happens automatically when a user enters the system and a filter exists for that user.

RELATED TOPICS

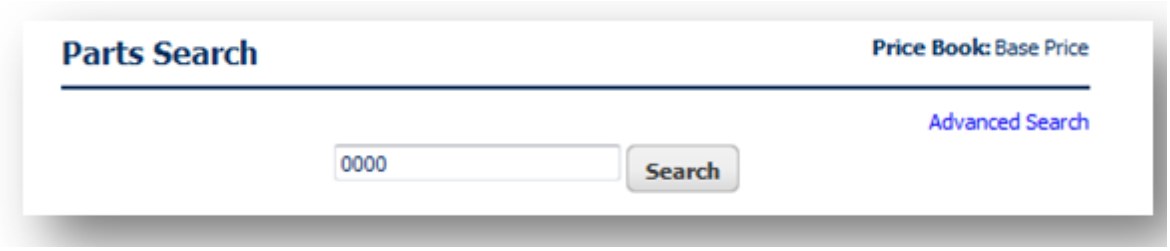
ADVANCED PARTS SEARCH

OVERVIEW

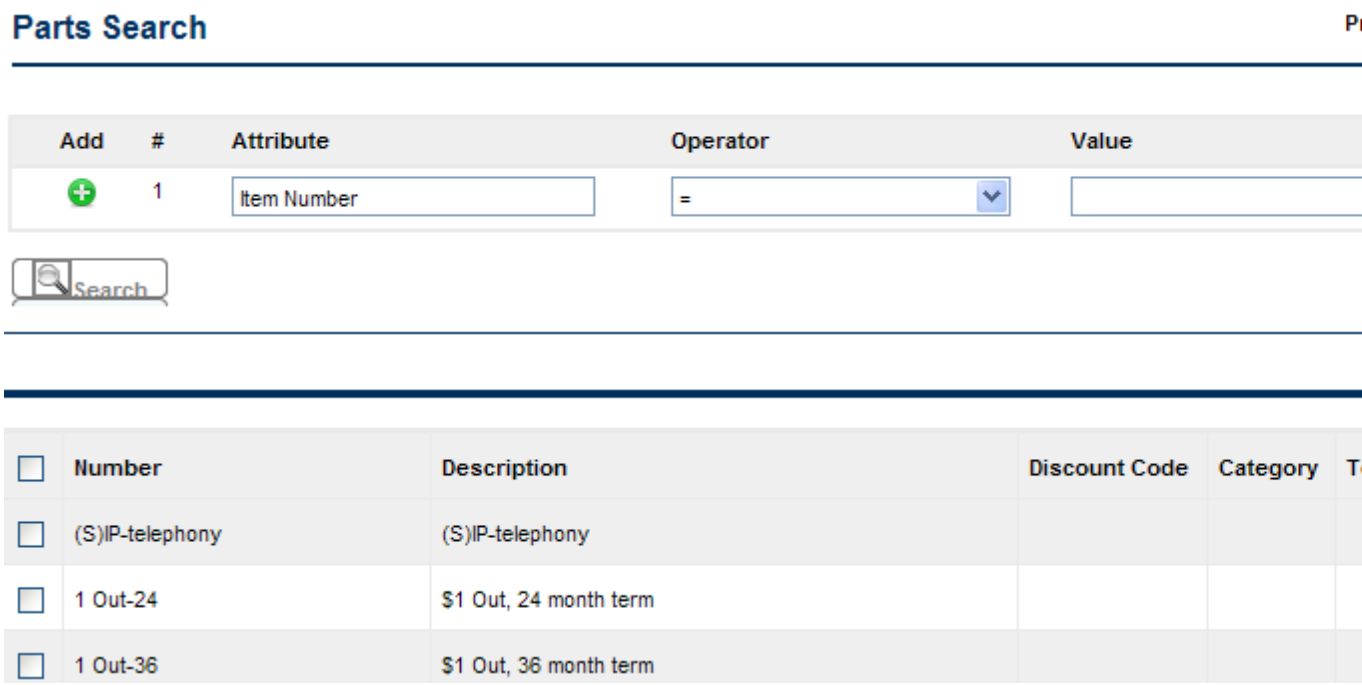
The Advanced Parts Search Editor is part of the parts search results page. The administrator can define custom XSL templates for both the Part Search Results panel and the Part Search Results section.

Searches that can be performed in the Parts Search Panel

- Simple



- Advanced



The Advanced Parts Search feature enables the end user to view their query and the parts search results on the same page. Users can search by Part Number, Description and all deployed custom parts fields. The user can create queries by stringing together conditions using 'AND' and 'OR'. The keyword search automatically performs a search on all search-able part fields using the 'contains' operator. In addition, the end user can make use of the row grouping feature to create more advanced queries.



There is a link that toggles back and forth between these two search types.

ADMINISTRATION

Setting up the Parts Search Panel

1. Navigate from: Admin > Page Templates > Parts Search Results Panel.
2. Select **XSL Templates** for both Production and Test.
3. Click the link under XSL Template (`parts_search_panel_xsl.jsp`).
4. Click **Yes** when prompted to save the page.
5. Upload the file `parts_search_panel.xsl` to both Production XSL File and Internal Test XSL File. *See attached file.*
6. Click **Update**.



If Parts Results Search Panel doesn't exist in Page Templates, log a case with OPS.

Parts Search Results

Here, you'll update the styling to match the Parts Search Panel.

1. Navigate from: Admin > General > General Options > Sticky Buttons on Configuration Page and Parts Search Results Page.
2. Click **Yes** on the radio button to enable the Stick Action Bar.
3. Navigate from: Admin > Styles and Templates > Page Templates > Parts Search Results Page.
4. Select XSL Template for both Production and Test.
5. Click the link under XSL Template.
6. Click **Yes** when prompted to save the page.
7. Upload the file `parts_search_results.xsl` to both Production and Internal XSL File. *See attached file.*
8. Click **Update**.

NOTES



Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.*

Upon upgrade, you must take the following steps to enable the Advanced Search button the home page and the Search panel on the Search Results page:

-To enable the Advanced Search Button on the home page, edit the home page XSL to construct the button.

-To enable the Search panel on the Search page, navigate to the Templates section and upload the Search Panel template in the template section. *A default template will be available for download in the same area.*



You can enter the search criteria in the Parts Search available on the home page. *If no results are returned, or multiple results are returned, then the user will be directed to the Advanced Parts Search page.*

The Parts Search Results page will have a sticky action bar. *You can turn this on in the General Options section of the Admin Home Page.*

When using wildcard searches that return a single part, the user will not be redirected to the part detail page. *Instead, the part will display on the part results page and the user can click on it to see the part detail page.*



Make sure that Shopping Cart is enabled. *This way you may continually add parts that are returned from different search results.*

RELATED TOPICS

EXPORT PARTS - BIGMACHINES AS PARTS MASTER

OVERVIEW

When updating the partner parts database

- If an attempt is made to export a part that has already been added to the Partner database at an earlier export, then that part is updated at the latest export.
- If a new part (a part that was created after the last export) is exported, then that part is added to the Partner parts database.

When updating the Partner parts database with Parts Extended Description

- Parts Extended Description Fields can be mapped to custom fields in Siebel or SFDC.
- The Parts database must be synchronized before Extended Description export is performed.
- If the size of the exported data is larger than the mapped field size in SFDC, the export data may be truncated.
- If the size of the exported data is larger than the mapped field size in Siebel, then an error is thrown.

When updating the Partner price books with an export of the price books (SFDC only)

- A price book in can be mapped to a price book in SFDC in a one-to-one relationship only.
- If a part that has a base price but is not associated with a price book is exported from, then that part will be associated with the Standard Price Book in SFDC.

ADMINISTRATION

Exporting Parts with BigMachines as the Parts Master

If the Partner system is SalesForce, then map the default price book to the SFDC standard price book. Go directly to Step 6 if the Partner system is Siebel.

1. Navigate from the Admin Home Page > Products > Parts
2. Click the **List** button in the Price Book section.
3. Click the **Name** of the required price book.
4. Fill in the **Partner Price Book ID** under the Integration section.



The Partner Price Book ID is available as a 15 character alphanumeric ID on the URL of that particular price book's page on Salesforce.com.

5. Repeat **Steps 1-4** to map the chosen price books with the corresponding SFDC price books.



Price books can only be mapped in as a one-to-one relationship.

6. Click on the **Integration** button to access the List Integrations page.



The SOAP Generator XSL and the Result Parser XSL must be loaded for both parts import and, if applicable, price book import.

7. In the List Integrations page, click on Name (for example, Import Parts and/or Import Price Book Entries).
8. Load the SOAP Generator XSL and the Result Parser XSL by clicking on Browse and loading the requisite files. Default SOAP Generator XSL and Result Parser XSLs are provided via links
9. Click on **Apply** to save changes.

Handling SFDC multi-select Attributes for mapping with corresponding Attributes

Use the template provided in the Result XSL, to map the multi-select attributes of SFDC with the corresponding attributes:

```
</xsl:template>
<!-- Template to replace charsIn(";") delimiter string with charsOut("~") delimiter string -->
<xsl:template name="replaceCharsInString">
<xsl:param name="stringIn"/>
<xsl:param name="charsIn"/>
<xsl:param name="charsOut"/>
<xsl:choose>
<xsl:when test="contains($stringIn,$charsIn)">
<xsl:value-of select="concat(substring-before($stringIn,$charsIn),$charsOut)"/>
<xsl:call-template name="replaceCharsInString">
<xsl:with-param name="stringIn" select="substring-after($stringIn,$charsIn)"/>
```

```
<xsl:with-param name="charsIn" select="$charsIn"/>
<xsl:with-param name="charsOut" select="$charsOut"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$stringIn"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

1. Select **Export Parts** and/or **Export Price Book Entries**.
2. Click on **Export** to invoke the import action.



You can also click on Schedule, in the List Integrations page, to access the scheduler. You can schedule integrations hourly, daily, weekly, monthly or On Demand.

NOTES



If users wish to export Pricebook Associations, they may do so separately.

Users may still import older Excel documents that contain Pricebook Associations columns.

Price Book Entries should be exported only after the parts are exported. If you to run a Price Book Entries export without running a parts export, then you must manually make sure that the parts databases are synchronized.



Parts Export doesn't include Pricebook Associations columns.



Parts Integration is not enabled for all applications. Please contact BigMachines OPS to enable.

RELATED TOPICS

IMPORT PARTS- PARTNER AS PARTS MASTER

OVERVIEW

When you Update the parts database in BigMachines from a partner:

- The parts database is deleted and then repopulated with the parts from the Partner Parts database.
- The price books are updated with the partner price books (applicable for SFDC only). A price book in BigMachines can be mapped to a price book in SFDC in a one-to-one relationship only.

ADMINISTRATION

Importing Parts with a Partner as the Parts Master

If the Partner system is SalesForce, then you will need to map the chosen price books to the corresponding SFDC price books. If you are using Siebel, please start at Step 6.

1. Navigate from the Admin Home Page > Products > Parts
2. Click on the **List** button in the Price Book section.
3. Click on the **name** of the required price book.
4. Fill in the **Partner Price Book ID** under the Integration section.



The Partner Price Book ID is available as a 15 character alphanumeric ID on the URL of that particular price book's page on Salesforce.com.

5. Repeat **Steps 1-4** to map the chosen price books with the corresponding SFDC price books.



Price books can only be mapped in as a one-to-one relationship.

6. Click on the **Integration** button to access the List Integrations page.



The SOAP Generator XSL and the Result Parser XSL must be loaded for both parts import and, if applicable, price book import.

7. In the List Integrations page, click on Name (for example, Import Parts and/or Import Price Book Entries).

8. Load the SOAP Generator XSL and the Result Parser XSL by clicking on Browse and loading the requisite files. Default SOAP Generator XSL and Result Parser XSLs are provided via links

9. Click on **Apply** to save changes.

Handling SFDC multi-select Attributes for mapping with corresponding Attributes

Use the template provided in the Result XSL, to map the multi-select attributes of SFDC with the corresponding attributes:

```
</xsl:template>
<!-- Template to replace charsIn(";") delimiter string with charsOut("~") delimiter string -->
<xsl:template name="replaceCharsInString">
<xsl:param name="stringIn"/>
<xsl:param name="charsIn"/>
<xsl:param name="charsOut"/>
<xsl:choose>
<xsl:when test="contains($stringIn,$charsIn)">
<xsl:value-of select="concat(substring-before($stringIn,$charsIn),$charsOut)"/>
<xsl:call-template name="replaceCharsInString">
<xsl:with-param name="stringIn" select="substring-after($stringIn,$charsIn)"/>
<xsl:with-param name="charsIn" select="$charsIn"/>
<xsl:with-param name="charsOut" select="$charsOut"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="$stringIn"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

1. Select **Import Parts** and/or **Import Price Book Entries**.

2. Click on **Import** to invoke the import action.



You can also click on Schedule, in the List Integrations page, to access the scheduler. You can schedule integrations hourly, daily, weekly, monthly or On Demand.

NOTES



If you run a price book entry import without running a parts import beforehand, then you must manually make sure that the parts databases are synchronized.



Price Book Entries/Parts Extended Descriptions are imported only after the parts are imported.



Parts Integration is not enabled for all sites. If you would want to integrate your site with either SalesForce or Siebel, please contact BigMachines Ops.

RELATED TOPICS

SERIAL NUMBERS AND ASSOCIATIONS

OVERVIEW

Adding a Serial Number

1. Navigate from Admin Home Page > Parts > Add New Serial Number.
2. Enter a unique Serial Number.
3. Provide a **Description** for the serial number.
4. Using **Browse**, select and add up to four files related to the serial number.
5. Provide values for any custom serial number fields
6. Click **Add**.
7. Click on a serial number to edit.



After creating a serial number, you can create it's BOM .

Deleting a Serial Number

Serial numbers can be deleted from the commerce system. When a serial number gets deleted, its BOM gets deleted too.

1. Navigate from Admin Home Page > Parts > Search for Part by Serial Number
2. On the Serial Number Search Results page, mark the checkbox corresponding to the serial number to be deleted.
3. Click Delete. The refreshed view of the Serial Number Search Results page shows the deletion.



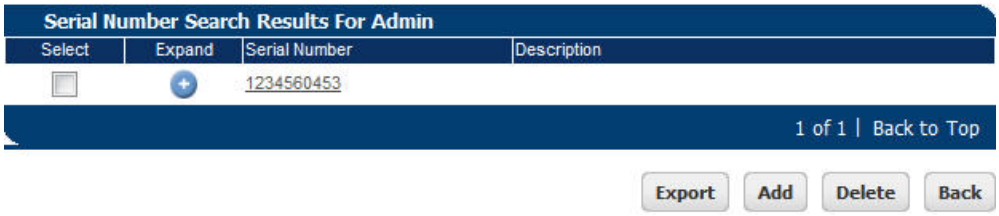
Use caution when deleting serial number, they cannot be recovered.

Associating a Serial Number

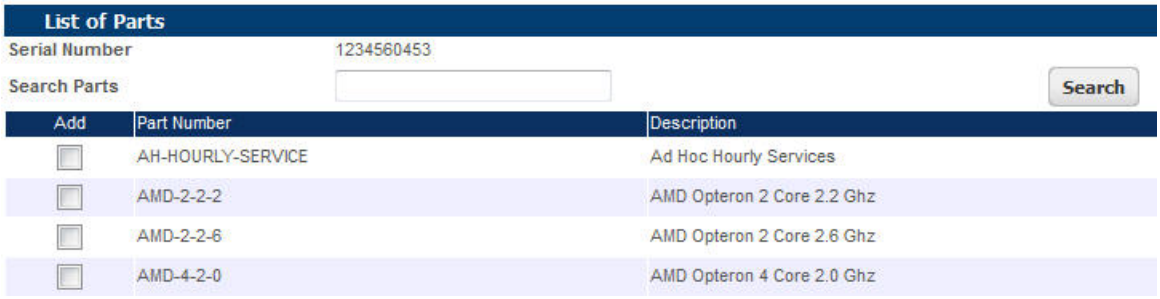
Serial number associations combine to form a BOM (Bill of Materials). BOMs can be created by associating parts with serial numbers. When parts are associated with child parts, then child parts are also included in the BOM.

1. Navigate from Admin Home Page > Parts > Search for Part by Serial Number
2. Search for the serial number whose association you want to create, or create a new serial number.

3. On the search results page, click the expand icon . This will take you to the Serial Number Associations page.



4. Click **Add**.
5. Search for the parts you want to associate.
6. Create associations by marking the checkboxes corresponding to the parts, and then click Associate.



Deleting Serial Number Associations

Serial number association deletions break the link between serial number and child part. Serial numbers do not get deleted in this process.

1. Follow the Standard Navigation Path and search for the serial number whose association you want to delete.
2. Click the expand icon . This will take you to the Serial Number Associations page.
3. Mark the checkbox corresponding to the part associations you want to delete
4. Click **Delete**.



It is not possible to undo a serial number association deletion.

Customizing Serial Number Fields

Customized Serial Number Fields are useful to assist in parts search as well as to set serial number filters.

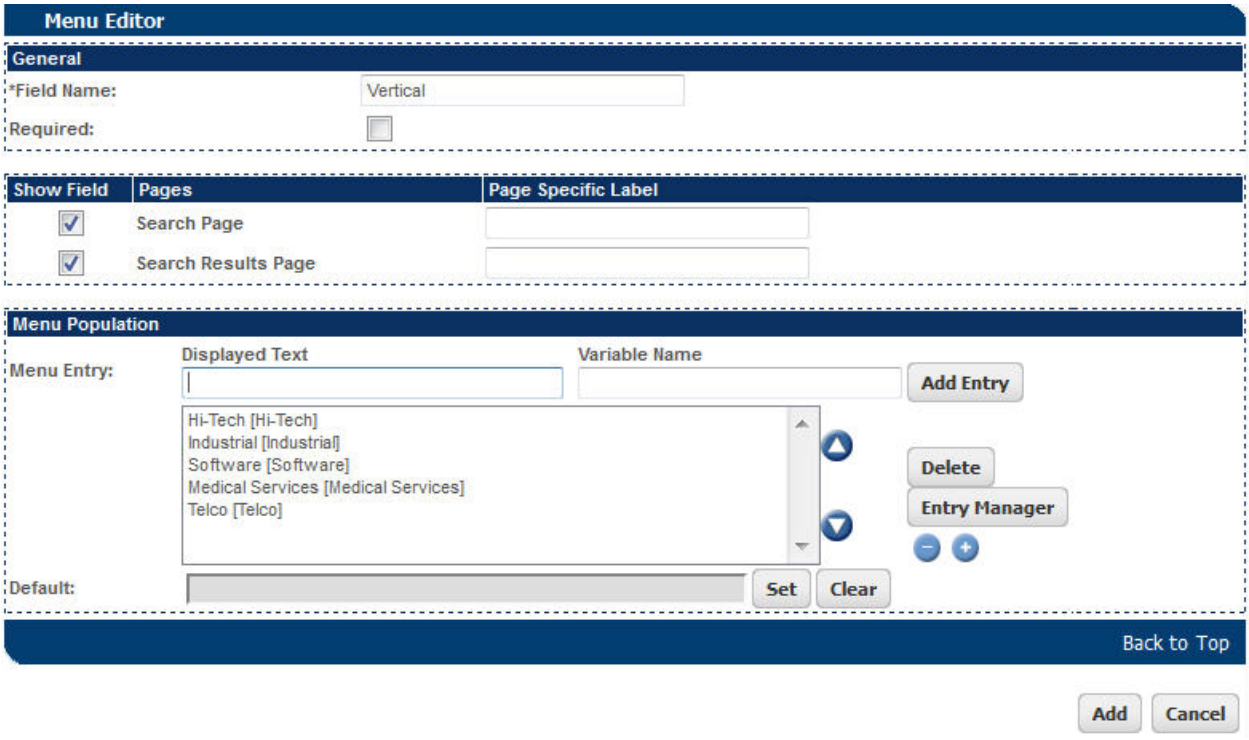
Pre-defined fields

There are 15 pre-defined fields available for specifying custom serial number fields:

- **Text:** There are 13 fields that can be set as String or Single Select Menu fields.
- **Number:** There are 2 fields that can be set as Float or Integer fields.

---

1. Standard Navigation Path: Admin Home Page > Parts > Customize Serial number Fields
2. Select either a **text or number field** and then choose the field type using the drop-down list provided.
3. Click on th**e Create Field** button.
4. Enter a **Field Name**.
5. Select **Required** if fullaccessusers must fill out this field when creating new serial numbers.
6. Select Search Page and Search Results page options to display the custom field in one ore both areas.
7. Enter menu values, if you've selected a Single Select Menu text option.



The Serial Number Defined Fields section lists fields that have already been added as custom fields. You can also specify if a field is going to be used as a Filter.

The Serial Number Available Fields section lists the fields that are available for creating custom fields.
If any fields have been defined, they will appear in the Defined Fields section.

The Serial Number Available Fields section lists the fields that are available for creating custom fields.
If any fields have been defined, they will appear in the Defined Fields section.

RELATED TOPICS

PARTNER PARTS INTEGRATION

OVERVIEW

The purpose of this article is to provide a high-level overview of partner parts integration. Partner Parts Integration allows you to maintain the primary parts database in either BigMachines or a Partner system. Currently, the Partner systems that can be integrated with to support Partner Parts Integration are:

- **SalesForce:** Salesforce.com maintains parts as ”r;products” associated with price books. Both parts and price book associations can be imported from or exported..
- **Siebel:** Parts can be successfully imported from or exported to Siebel. The import/export Price Book Entries option is disabled in case of Siebel as Siebel does not have price books.

ADMINISTRATION

Export

When the Parts Master is BigMachines, the parts/price books are maintained in your BigMachines application . Hence parts and /or price book associations have to be exported to the Partner system from your BigMachines environment. Parts extended descriptions can also be exported from BigMachines to the Partner system.

Import

When the Parts Master is the Partner system, parts/price books are maintained in the partner system. Hence parts and/or price books have to be imported from the Partner system into BigMachines. Parts extended descriptions can also be imported from the Partner system to BigMachines.

Handling SFEC multi-select Attributes for mapping with corresponding Attributes

Salesforce.com(SFDC) stores multi-select Menu related values using (";") in its format while BigMachines stores its Multi-Select Menu attribute values using the tilde delimited format("~"). For example, "AAA~BBB~CCC".

Therefore, while importing parts into BigMachines, multi-select menu data from SFDC should be converted to the tilde delimited format. Similarly, while exporting Parts from BigMachines into SFDC, the multi-select menu data in BigMachines needs to be converted into the semi-colon delimited format for SFDC.

NOTES



Only the fullaccessuser is capable of invoking an Integration (Export/Import).



Parts Integration is not enabled for all sites. If you would want to integrate your site with either SalesForce or Siebel, please contact the System fullaccessuser.

RELATED TOPICS

PARTS- USE CASES & TROUBLESHOOTING

USE CASES

Handling discontinued parts

This Use Case outlines how to appropriately handle an existing quote that contains a discontinued part/product.

1. Use a custom field for the **End Date** of the product.
2. Construct a warning message (in Commerce) that appears when the user adds a discontinued part/product to the quote. *This highlights that the product has an **End Date** that precedes the current date/quote expiration date.*
3. When the user submits a quote for approval with an expired product, **EITHER**:
    a. Create a Validation Rule that stops users from submitting the quote altogether.
    b. Create an “Expired Product” approval trigger.
4. Use similar logic when a user attempts to place an order that contains an expired product/part. *Either block the user from placing an order with an expired product or require approval.*

TROUBLESHOOTING

Buttons missing on Parts Integration page

The Parts Integration can only be run by superuser. *If you are logged in as superuser, then these actions will only be visible if you are punched into a Partner site.*

- If your organization is able to punch into your integrated site from BMI:
    a. Click the **Partner Login** link.
    b. Refresh the page.
- If your organization is NOT able to punch into your integrated site from BMI:
    a. Return to the partner site.
    b. Punch back into BMI.
    c. Navigate to Parts Integration.



If the "Partner Login" link is still visible on the top of your screen, you are not punched in.

RELATED TOPICS

DEFINING PRODUCT SORT BEHAVIOR

OVERVIEW

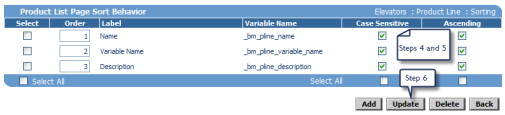
By default, items appear on the product line and model search results in order based on their internal system ID. You can change this by designating attributes you prefer the system to sort products by. To display products in as random an order as possible, create a sort order with attributes whose values fall in a narrow range.

ADMINISTRATION

1. Navigate from: Product Line Attribute List page > List Ordering.
2. Click **Add** to open the Add Product Attributes for Sorting page.
3. Select one or more attributes in the **Unsorted Attributes menu** and click **Add**.



4. Select the **Case Sensitive** checkbox corresponding to text attribute types.
5. Select the checkbox under the **Ascending** column.
6. Click Update to save changes to the page and click Back to return to the Product Attribute List page.



NOTES

To stop using a sort attribute, Select the checkboxes corresponding to the attributes you no longer want to use for product sorting and click Delete.

RELATED TOPICS

USER ADMINISTRATION

OVERVIEW

Anyone that you want to conduct business on your company's behalf must have a user account. This page displays information and options related to the internal users.

User Admin List page abilities

- **Add User Accounts**: Click Add to add user accounts and group associations from the User Administration page.
- **Edit User Accounts**: Click a Login ID to edit user accounts and group associations using the User Administration page.
- **Activate/Inactivate Users**: Use Select/Select All to select a user's login ID and click Inactivate to inactivate the required users.
- **Sort fields:** Sort users based when you click on the Login, User Name, User Type or Partner Login label.
- **Export User Details:** Export all user details in the CSV format. Note: You will see the Exportoption only if your system supports integration.
- **Activate or Deactivate Users:** Use the Select/Select All option and click Activate to activate a few or all of the inactive users.

Proxy Login

When you click the proxy login ID against a user type - Full Access, Channel Agent, Sales Agent, or Restricted Access - you are logged into the system as that particular type of user, with the access rights that the user enjoys. You can do anything that the particular user type selected can do. Clicking the Proxy Profile link displays the User Administration page for that user. Remember, if you click Logout, you will be logged out of the system - not just from the proxy login of that user type.

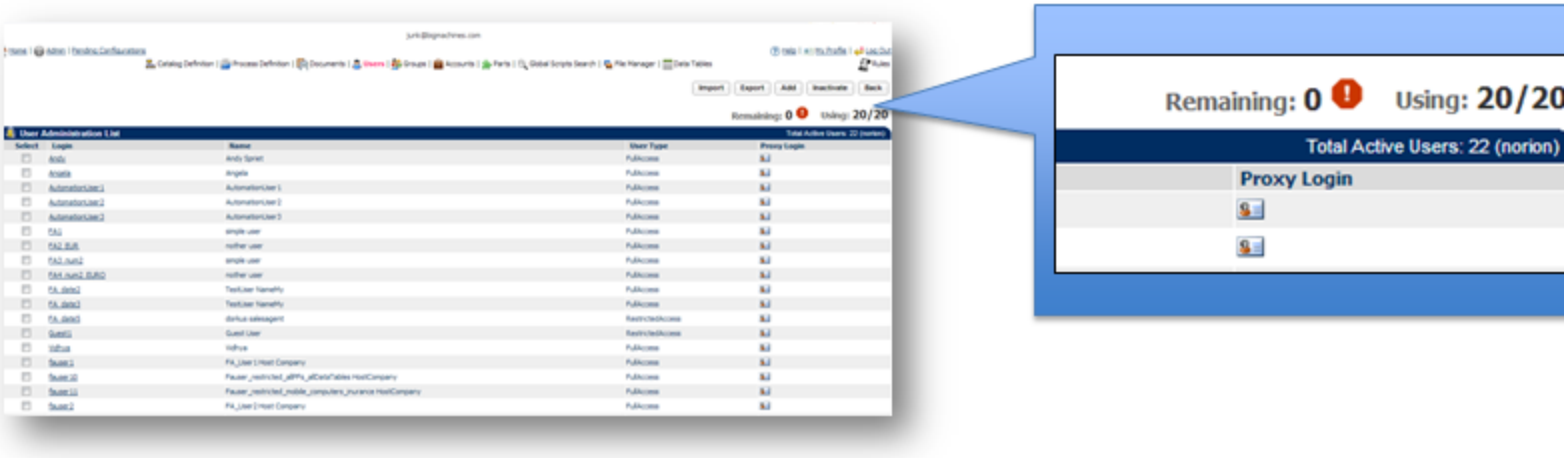
User License Count

There is an indicator that informs you of how many purchased user licenses remain available for use. This feature is available within Internal Users and Partner Organization administration pages.

	NEW FEATURE DISPLAYED TO ADMIN		
	Internal Users Page	Partner Org Internal Page	Partner Org User Admin Page
HOST COMPANY ADMIN	YES	YES	YES
PARTNER ORG ADMIN	NO	NO	YES

System-generated users (superuser and guest) are included in the Total Active Users, but are not included in the active licensed user count of "Remaining" and "Active" users. Notice in the image above that there are **22 Total Active Users**, but the **Using** licensed user count remains at 20. These two additional users and the "SuperUser" and "Guest" account.

A red stop sign icon appears next to the **Remaining** licensed user count.when there are no further remaining users. An orange warning icon appears when user capacity reaches less than 5%.



The value in the parentheses indicates how many licensed users each Partner Organization has active.

NOTES

If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit. *When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.*

RELATED TOPICS

SETTING UP USERS

OVERVIEW

Internal users are usually full admin users, internal sales reps, managers, etc within the Host Company. Typically, distributors or other partner organizations are not defined in this area. The number of internal users is defined by the BigMachines Customer Agreement.

Only the superuser and fullaccessusers within the Host Company can view and edit all Host Company information, accounts and internal users. They can also create, view and edit the accounts of all Partner Organizations and the user associated with these organizations.

User Type Permissions

FUNCTION	SUPERUSER	FULLACCESSUSER	SALESAGENT USER	CHANNELAGENT USER	RESTRICTEDACCESS USER
Archive Commerce	X				
Upload Products	X	X			
Modify Commerce	X	X			
Modify Configuration	X	X			
Perform Migration	X	X			
View/Modify Users	X	With Permission			
Maximum Users field is visible on each Partner Organization's Company Administration page.	X	With Permission			
Manage Accounts	X	X	X	X	
Account Associations	X	X			
Manage Groups	X				
Manage Personal User Information	X	X	X	X	X

Table 1.1: User access to administration

By checking the "Permission to create/modify users" box, fullaccessusers are given some superuser rights, which includes the ability to proxy in as other users, modify users, reset passwords, add new users, perform a commerce archive, schedule usage reports and create shared folders.

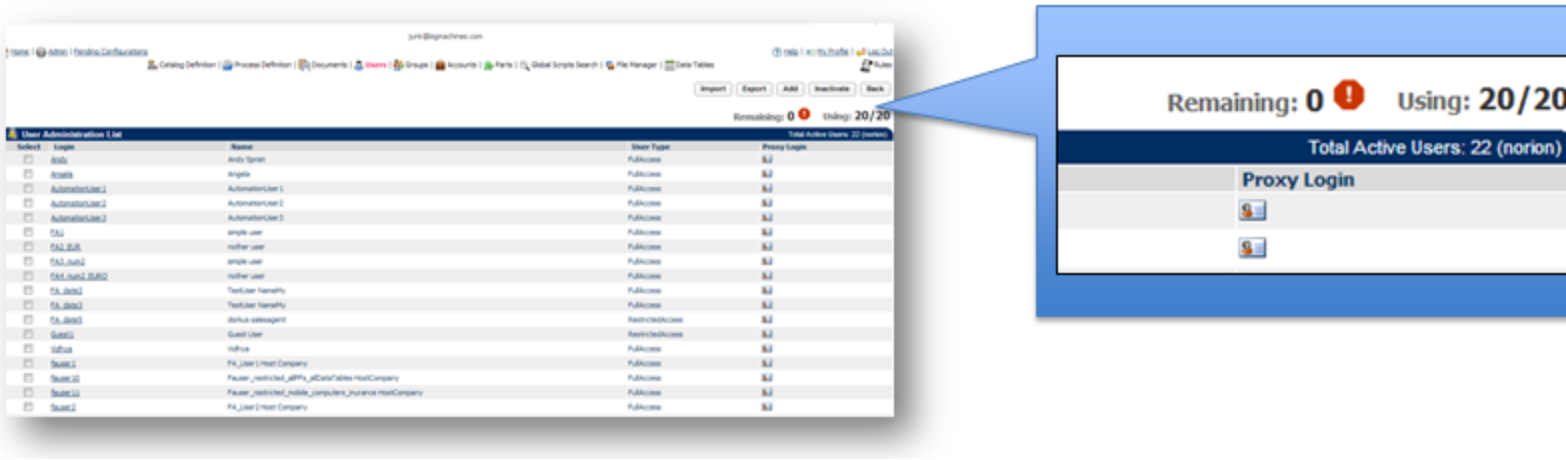
User License Count

There is an indicator that informs you of how many purchased user licenses remain available for use. This feature is available within Internal Users and Partner Organization administration pages.

	NEW FEATURE DISPLAYED TO ADMIN		
	Internal Users Page	Partner Org Internal Page	Partner Org User Admin Page
HOST COMPANY ADMIN	YES	YES	YES
PARTNER ORG ADMIN	NO	NO	YES

System-generated users (superuser and guest) are included in the Total Active Users, but are not included in the active licensed user count of “Remaining” and “Active” users. Notice in the image above that there are **22 Total Active Users**, but the **Using** licensed user count remains at 20. These two additional users and the “SuperUser” and “Guest” account.

A red stop sign icon appears next to the **Remaining** licensed user count.when there are no further remaining users. An orange warning icon appears when user capacity reaches less than 5%.



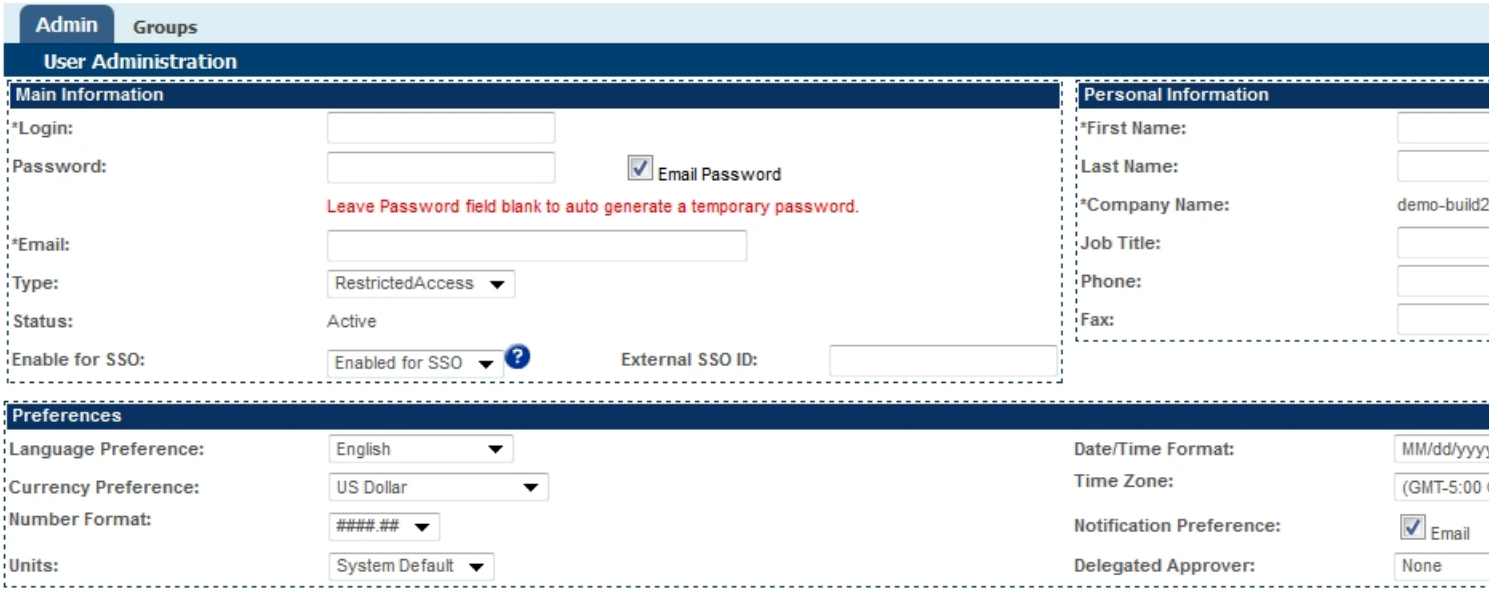
The value in the parentheses indicates how many licensed users each Partner Organization has active.

ADMINISTRATION

Setting up Internal Users

We recommend using the bulk upload process for entering large amounts of user data. For small amounts of data, manual entry may be a quicker process.

1. Navigate from Admin Home Page > Users > Internal Users.
2. Click **Add** to open the User Administration page.
3. Enter the login, email, user type, and first name of the user. *Note: This is the minimum required information*
4. Select a **User Type**. *See User Type Permissions table.*



5. Click the **Groups** tab to add the user to a group.
6. Click **Add** to save the user.

NOTES

Clicking My Profile displays information related to the logged in user.

Periods can be used in BigMachines usernames.

Only superuser or fullaccessusers can view or modify other user profiles. All lower access users can only view and modify their own profile.

Administrative functions can only be performed by fullaccessusers, including superuser. Admin functions include making changes to configuration (adding attributes, creating rules, etc.) and modifying commerce processes.

Once created, users cannot be deleted. You can inactivate a user instead.



If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit. *When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.*

RELATED TOPICS

SETTING UP GROUPS

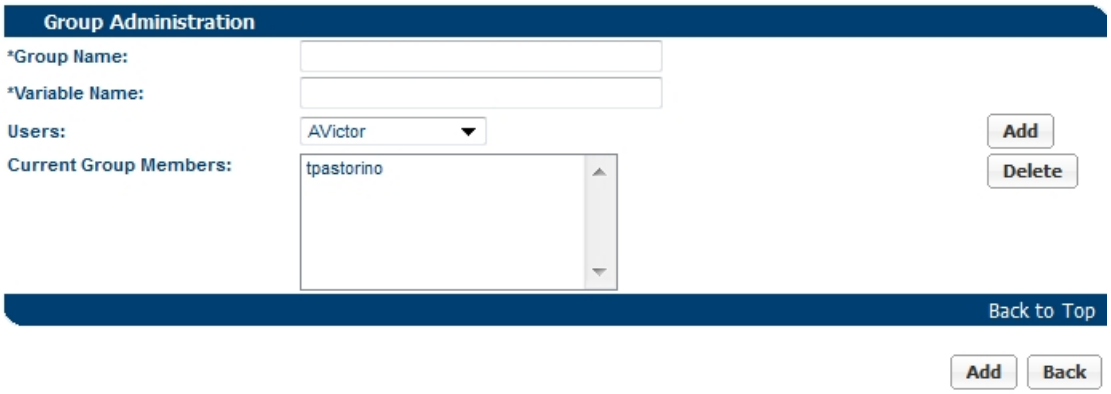
OVERVIEW

Groups are a collection of users and are useful for determining access rights for commerce documents. Once you land on the Group Administration List page, you can Add or Edit user groups.

ADMINISTRATION

Setting up Groups

1. Navigate from the Admin Home Page > Groups.
2. Click **Add**.
3. Enter a **Group Name** and **Variable Name**. *When assigning variable names, avoid non-alphanumeric characters and spaces.*
4. Add to the group by choosing users from the Users drop-down menu.
5. Click **Add**.



Once you have added your users, click Add or Update to save the added information or click Back to go back to the Group Administration List page.

A user can belong to more than one group. Groups can be managed only by the SuperUser and FullAccessUser.

Maintaining a Group

1. Edit the group name by modifying the existing name in the Group Name field, and click Update.
2. Add users by selecting a user name from the Users drop-down list, and click Add. The added users can be seen in the Current Group Members field.
3. Remove users by selecting one or more users in the Current Group Members field, and click Delete.
4. Save the changes made by clicking Update or click Back to go back to the Group Administration List page.

NOTES



Periods can be used in BigMachines usernames.

Groups can not be deleted once they are added. If you no longer need a group, just change the groups name to something like "***DELETE ME***". This will let the admin users know that this group is no longer used.



Determining what groups to have is decided at the time of implementation design. Once a site has been implemented, there is usually no need to add additional groups unless new functionality is required. Groups are typically used for commerce process workflow permissions, although you can also decide what a user will see (or not see) on the quote form, depending on their group.



If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit. *When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.*

RELATED TOPICS

HOST COMPANY

OVERVIEW

The Host Company refers to the hosting company, which is of FullAccesswithESales company type.

Host Company user types

- FullAccess
- ChannelAgent
- SalesAgent
- Restricted Access

Only the SuperUser and FullAccess users within the Host Company can view and edit all Host Company information, accounts and internal users. They can also create, view and edit the accounts of all Partner Organizations and the users associated with these organizations.

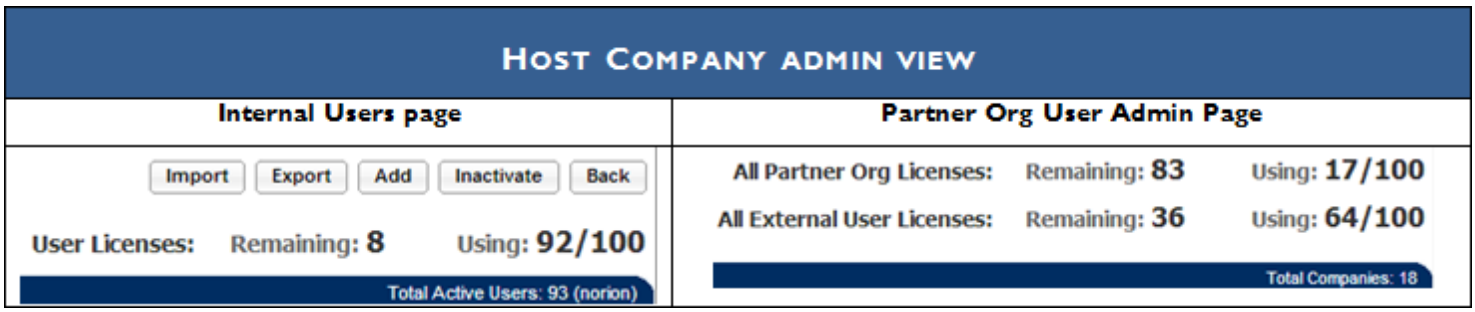
ADMINISTRATION

Managing Host Company Information

1. Navigate from Admin Home Page > General > Host Company
2. Edit the Host Company name, email, and address details.
3. Select from a list of Supported Languages. To edit this list, please contact BigMachines Ops.
4. Edit the list of supported currencies. *To edit this list, please contact BigMachines OPS.*
5. Edit the number format, units, date/time format and time zone.
6. When finished, click Update. Doing so refreshes the content and saves it for future reference.

User License Count

There is an indicator that informs you of how many purchased user licenses remain available for use. This feature is available within Internal Users and Partner Organization administration pages.



The Host admin has the most comprehensive view of the new license information provided. The fullaccessuser or superuser will see:

- Purchased Internal User licenses (“User Licenses”)
- Purchased Partner Organization licenses (“All Partner Org Licenses”)
- Purchased External User licenses distributed across all Partner Orgs (“All External User Licenses”)
    - “Users” column has become “Users (Active Licensed Users)” column.
- The value in the parentheses indicates how many licensed users each Partner Organization has active.
    - The “Maximum Users” field is now also visible to host “FullAccess” user with “SuperUser” permissions on each Partner Organization’s Company Administration page.
    - Legacy users that renamed their “guest” will notice that an additional “guest” user has been created to replace the renamed “guest”.
- The renamed “guest” will become an “Active Licensed User” and be counted against the licenses totals.
- The newly added “guest” user is inactive.

System-generated users (superuser and guest) are included in the Total Active Users, but are not included in the active licensed user count of “Remaining” and “Active” users.

A red stop sign icon appears next to the **Remaining** licensed user count.when there are no further remaining users. An orange warning icon appears when user capacity reaches less than 5%.

NOTES



Periods can be used in BigMachines usernames.



If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit. *When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.*

RELATED TOPICS

PARTNER ORGANIZATIONS

OVERVIEW

Partner Organizations are typically used when a BigMachines customer has independent third parties that quote and/or sell products (typically resellers, distributors or VARS). One BigMachines instance can be enabled for both standard and partner sales users.

All partner organization accounts are created as BuyAccess company types. (Your company is a FullAccessWithESales company type.) The superuser of each partner organizations account can manage its own users, groups and company information by logging in to your system with the appropriate account information. After creating partner organization accounts, make sure to distribute account information to the appropriate fullaccessusers for each company.

User License Count

There is an indicator that informs administrators of how many purchased user licenses remain available for use. The Partner Organization administrator has a limited view. They will be able to see the **Using** license count used by that particular Partner Organization. The only way a Partner Organization admin can see their **Remaining** licenses is if the host set a value in the **Maximum Users** column for ALL the Partner Organizations.

If a Host Company superuser, or fullaccessuser with permissions, has set a **Maximum Users** value for all listed Partner Organizations, then each Partner Organization administrator will see their "Remaining" user license count and the "Using" ratio for active licenses consumed/total licenses remaining on their Users page.

If a Partner Organization with Denied access does not have a value set by the Host admin in the Maximum Users field, no Partner Organization admin will see their "Remaining" information.

PARTNER ORGANIZATION ADMIN VIEW OF USERS PAGE (INCLUDES "DENIED" PARTNER ORGS)	
Values set by Host superuser	**Values NOT set by Host superuser**
	

Known Issues

- "Denied" Access Partner Organizations must set their "Maximum Users" value to zero.
- If the original "Maximum Users" value (when Access was still "Granted") is kept, it will count against the Host admin's "Remaining" total for all Partner Organizations.
- Additionally, if the value is not set to zero, no other Partner Organization will be able to see their "Remaining" licenses value for their organization.



To allow “Granted” Partner Organization admins to view their “Remaining” license information, Host admin must enter a zero in the “Maximum Users” column for any “Denied” Partner Organization.

ADMINISTRATION

Setting up a Partner Organization

1. Navigate from the Admin Home Page > Users > Partner Organizations.
2. Click **Add** on the Company Administration List page.
3. Enter a **Full Company Name** and a **Company Login**.



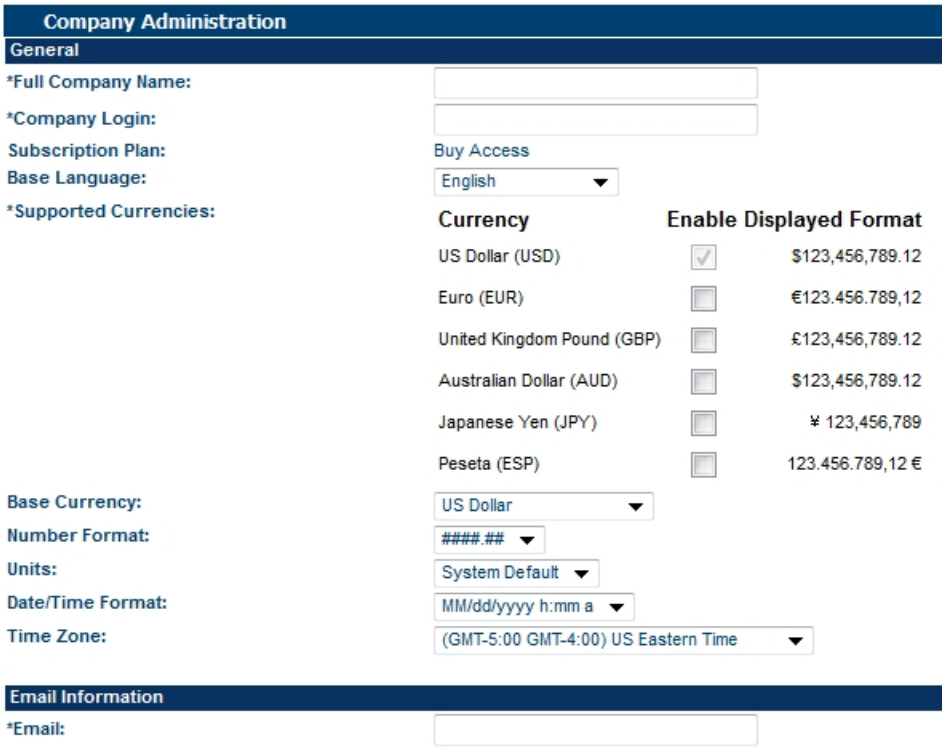
Company Login is where you give your company a login name. This will append the login for all users, so best practice is to keep it simple. A sample login is "JoeSmith@JoesCompanyName"

4. Choose a **Base Language** from the drop-down menu, then select supported currencies.



The partner organization's administration cannot set its own supported currencies. Before creating a new partner organization account, determine the currencies the company needs to support so that they can be accurately set-up at the time the account is created.

5. Choose options for base currency, number format, units, date/time format and time zone.
6. Enter a company email address. *The email for the company should be the person who will be designated as the superuser.*



7. Click **Add**.

8. **After you create a company account, you will receive login information:**

    a. The full access login account is always superuser@companylogin.and the password is always initialized as "superuser".

    b. Upon initial login, please make sure to change your password from "superuser".

Setting up Users in a Partner Organization

This page displays information and options related to the users of the selected Company. You can add, edit and inactivate users. You can also export company details.

Anyone that you want to conduct business on your company's behalf must have a user account. New user accounts can only be created by your company's superuser. We recommend using the bulk upload process for entering large amounts of user data. For small amounts of data, manual entry may be a quicker process.

FUNCTION	SUPERUSER	FULLACCESSUSER	BUYACCESS USER	RESTRICTEDACCESS USER
Upload Products	X	X		
Modify Commerce	X	X		
Modify Conifugration	X	X		
Perform Migration	X	X		

FUNCTION	SUPERUSER	FULLACCESSUSER	BUYACCESS USER	RESTRICTEDACCESS USER
View/Modify Users	X			
Manage Accounts	X	X	X	
Manage Groups	X			
Manage Personal User Information	X	X	X	X

Table 1.1: User access to administration

1. Navigate from: Admin Home Page > Users> Partner Organizations > Choose a Company > Users > List.
2. Click **Add** on the User Administration List page.
3. Enter the login, email, user type, and first name of the user. *Note: This is the minimum required information*. You can also manage the following:
    - Addition personal information (last name, job title, etc.)
    - Language and currency preferences
    - Preferred units and number format
    - Date/Time Zone
    - Notification preference
    - Delegated approvers
    - Bill To/Ship To Information
4. Click **Group** to add the user to a group
5. Click **Add** to save the user.

Customizing Commerce Views for Partner Organizations

A typical requirement is to customize the document view for partner users, for example, by hiding certain fields. This can be achieved using hiding rules or by using separate Participant Profiles for partners, if complex customizations are required.

Identifying Partner Users & Quotes

- **Identifying quotes created by a partner user:** create a quote-level `creatorCompany_quote` attribute, and default this value to the Current User's Company name system attribute (`_system_company_name`).
- **Identifying partner quotes:** compare the `creatorCompany_quote` to the Supplier's Company name system attribute (`_system_supplier_company_name`).
- **Identifying partner users:** compare the Current User's Company name system attribute (`_system_company_name`) to the Supplier's Company name system attribute(`_system_supplier_company_name`).

Creating Separate Participant Profiles

Creating separate Participant Profiles for partner users requires a combination of additional Steps and Participant Profiles. **Solution:** create 2 **Pending** steps:

- 1 partner quote
- 1 regular quote



In other steps, a single step with separate profiles can be used.



You can not create a new Participant Profile in the Pending Step and use the "User Access Rights" feature to assign it to a particular group representing partner users, because assigning a Participant Profile to a group gives access to all users in that group, not just the quote creator.

Similarly, it is not suitable to assign partner participant profiles based on User Type Access Rights (Full Access, Buy Access, etc). *This would make quotes visible to all users with that access type, meaning that Full Access users in one partner organization would see quotes from a separate partner organization.*

Partner vs. Internal Profiles

Steps	User Access Rights	Transition Rules
Start	• Partner Profile: Company type- Buy Access • Internal Profile: Otherwise	• If quote created by partner, transition to Pending – Partner. • If quote created by internal user, transition to Pending – Internal.
Pending	• Partner Profile: Performer Step (Start) • Internal Profile: Performer Step (Start)	Standard Transition Rules to Submitted for Approval or Approved
Submitted for Approval	• Partner Profile: Performer Step (Pending- Partner) • Internal Profile: Performer Step (Pending- Internal)	• Standard Transition Rules to Approved. • Revise or Reject must transition to Pending – Partner if quote created by partner user.
Approved	• Partner Profile: Performer Step (Pending- Partner) • Internal Profile: Performer Step (Pending- Internal)	Revise must transition to Pending – Partner if quote created by partner user.

Customizing Workflow

A common requirement is to introduce a new, first level of approval for partner quotes. **Example:** each partner sales rep may be assigned to an internal sales rep, and that internal rep may need to approve the partner's quote before it continues through the usual approval process.



For more information on the Commerce Workflow, see Workflow Steps.

Customizing Config, Parts, and Price Books

Create a data table to store Partner Organization variable names and the rules defining what Price Books, Parts (based on an appropraite custom field, like Product Line), and Configurators the partner can see. **For example:**

Partner Organization	Price Books	Product Lines	Configurators
ACME	Canada, US	Doors, Windows	DoorsAndWindows

Hiding Configurators

1. Navigate from: Admin > Home Page.
2. Located the appropriate Punch-in
3. Click the **Restrict Access** icon
4. Create a rule that returns **True** if the configurator should be hidden.

Hiding Parts

1. Navigate from: Admin > Parts > Customize Parts Fields.
2. Set Filter / Normal to Filter for the appropriate field.
3. Create an “Advanced Script” that returns a comma-delimited list of the custom field values for parts the user should be able to see (e.g., if the user should be able to see parts with the product lines Doors and Windows, return “Doors,Windows”).



Caution should be used to ensure that parts that are filtered from view are not used in Recommended Item rules.

Hiding Price Books

1. Navigate from: Admin > Parts > Price Book > Customize Price Book Fields.
2. Create a custom field and set Filter / Normal to Filter.
3. Store a unique identifier, such as the Price Book variable name, in the custom field for each Price Book. *This is required because you cannot filter the Price Book by variable name directly.*
4. Create an “Advanced Script” that returns a comma-delimited list of the custom field values for Price Books the user should be able to see (e.g., if the user should be able to use the Canada and US Price Books, return “canada,us”).

Caution should be used to ensure that parts that are filtered from view are not used in Recommended Item rules.

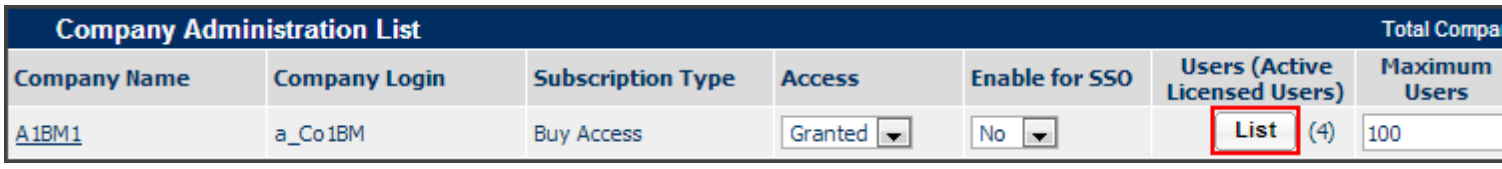
Once created, a partner organization can't be deleted.

A user can't be moved from one partner organization to another.

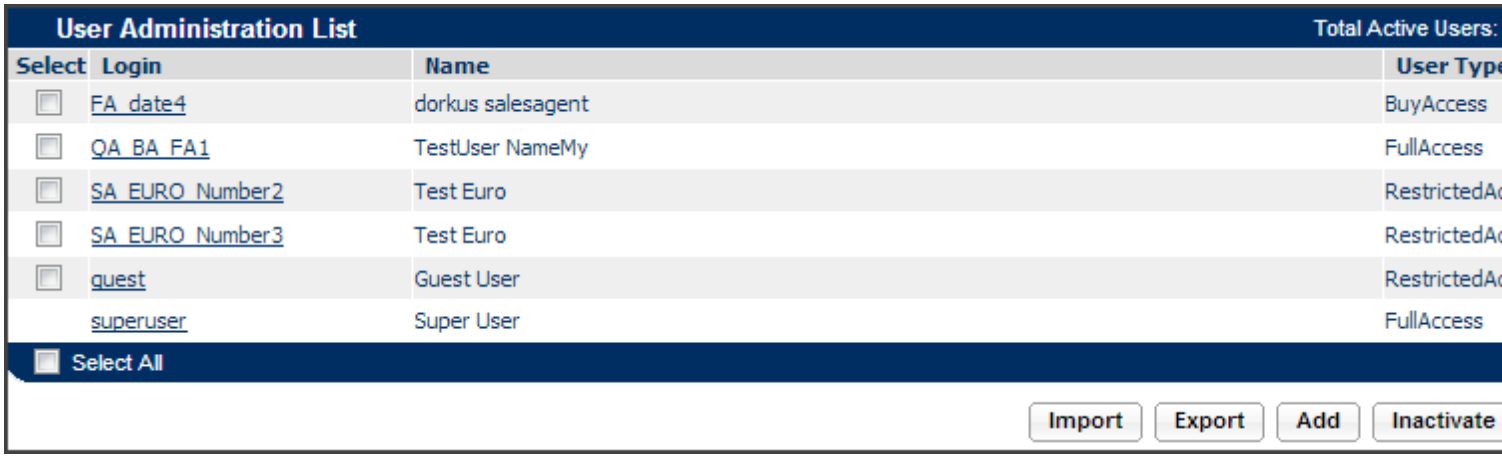
Partner users can't use the native “Forward” feature to forward a quote to a user in a different organization (another partner organization or the parent organization).

Changing Partner Org user password

1. Navigate from: Admin > Users > Partner Organizations.
2. Click the **List** button of the desired Partner Org company name.



3. Select the desired user **Login** name.



4. Click **Reset Password**. *This will email the user a new password.*



You must be logged in as a SuperUser to change a Partner Org user password.

TROUBLESHOOTING

Salesforce Integration Troubleshooting

The Salesforce partner management portals are: PRM and Communities. They should be used in cases where Salesforce integration is required for partner users. If partner users do not require Salesforce access, they can log in to BigMachines directly; however, access to accounts and opportunities will not be available, and partner-created quotes will not appear in Salesforce.

Problem	Possible Cause
Users can see quotes created by other users in the same partner organization.	Participant Profiles were assigned to groups on the User Access Rights tab.
Users can see quotes created by users in other partner organizations.	Participant Profiles were assigned to User Types on the User Access Rights tab.

NOTES



Currency preference changes take effect after the user has logged out and then logged back in.

Distributors are typically defined as Partner Organizations.

Periods can be used in BigMachines usernames.



By default, a company account called ”yourcompany_buy” also exists in the partner organization administration area. This company enables you to support quick registration. All users who submit a quick registration form get a login account ending with the extension ”yourcompany_buy”. If you do not want to support quick registration, the QuickRegistrationBuy company account should be denied access. To support quick registration, grant access to the company named, "YourCompany_buy".



It is not possible to delete a Partner Organization. To prohibit a company from accessing your system, toggle to the company's access rights and change it to Denied. Access rights take effect immediately after clicking Update Access.

It is not possible to delete a user account. To prevent a user from logging in, change their login information, for example, user name and password.

If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit. *When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.*

RELATED TOPICS

ADMIN ACCESS CONTROL

OVERVIEW

Access Control allows the superuser, or a fullaccsesuser with permission, to grant controlled access across various admin sections to a fullaccessuser type (without super user privileges). Administration permissions can be set up for product families and data tables (by folder) only.

By default, all Host Company fullaccessusers are given the permission to view/edit all administration modules (except users). When a fullaccessuser is restricted from any Product Family and/or data table, the user is barred from other sections of the Administration Platform. A restricted fullaccessuser therefore, can only access the Product family and/or data tables that she has access to.

ADMINISTRATION

Setting permissions

1. Login as **SuperUser or FullAccessUser** with superuser permissions.
2. Navigate from the Admin Home Page > Users
3. Choose a **FullAccessUser**.
4. Select the **Product Families** and/or **Data Tables** that this user will have access to from the Access Permissions tab.

NOTES



If you are considering using this, you should modify the navigation menu links to make it harder for restricted admins to access the areas from which they are supposed to be restricted.



A Restricted Admin User will have no access to other areas of the administration platform: If a full access user is restricted from any product family or data table (by folder), the user will automatically lose access to other areas of the Administration Platform. He will only have access to the Product Family and/or data table that the superuser has allowed him access to.



Admin Access Control does not impact the user side: This enhancement does not impact the user side in any way. A fullaccess user who is restricted from a product family can still interact with this product family on the user side if the implementation allows him to do so.

If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit. *When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.*

RELATED TOPICS

USER ACCESS RIGHTS

OVERVIEW

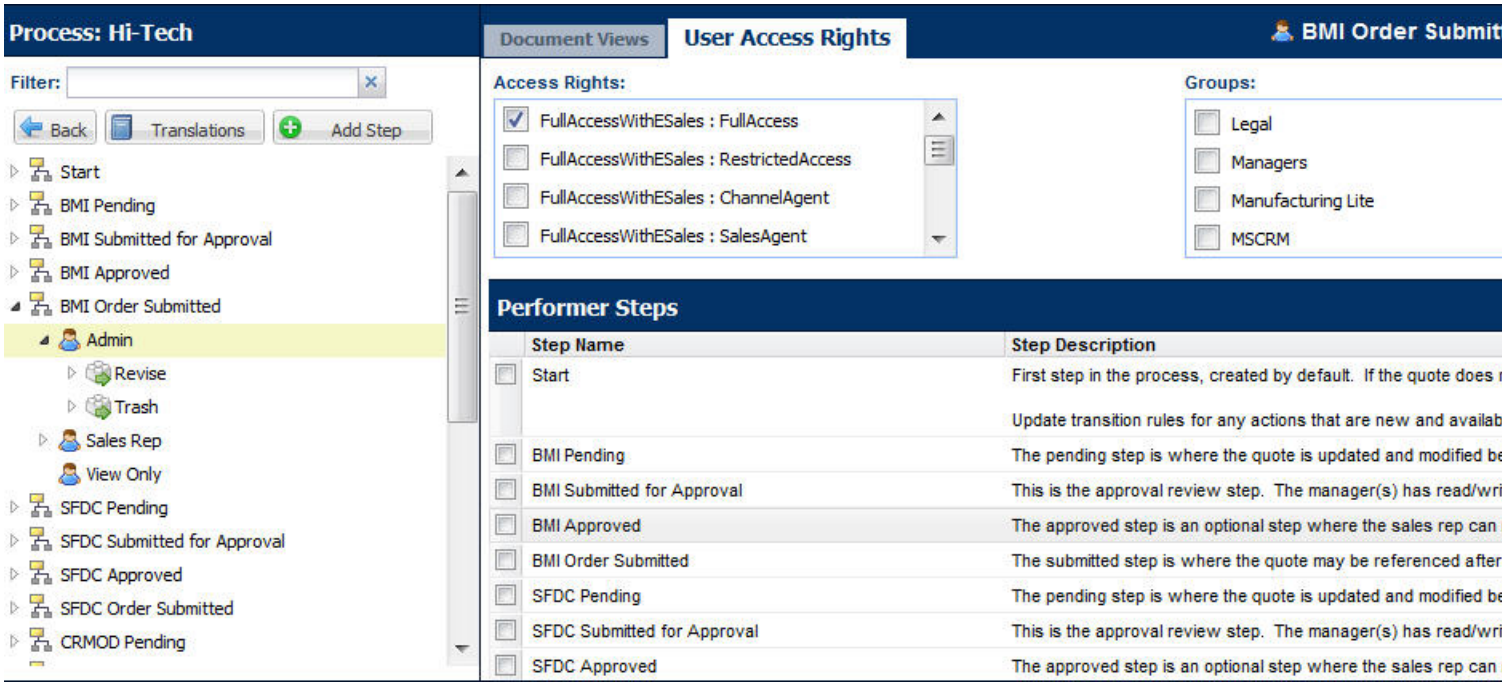
User access rights designate who is included in a profile. Access can be granted based on user type, group, or performers in other document steps. For more information on the user types available, see the User types reference. A user is included in a profile if one of the following conditions is satisfied:

- The user belongs to a company type and user type granted access rights
- The user belongs to a group granted access rights
- The user is a performer of any action in a listed step

ADMINISTRATION

Setting up User Access Rights

1. Navigate from: Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps
2. Click on a **desired step** to expand it.
3. Double-click the **Participant Profile** .
4. Click on the **User Access Rights** Tab.
5. Select one or more **user types** from the Access Rights menu.
6. Select one ore more **Groups** to add to the profile.
7. Select any step whose users you want to grant access to in the **Performer Steps** section.



8. Click **Save**.

"Web Services Only" user permissions checkbox

A “Web Services Only” checkbox is located on the Permissions section of the User Administration page. Only FullAccess users with the ability to create/modify users can change this setting. When the “Web Services Only” checkbox is selected for an internal user, that user may only make Web Services calls to the BigMachines site; logging in through the web interface will not be permitted.

NOTES

User can be granted access to profiles through Auto-Forwarding Rules.

You can remove a Company Type: User Type set from the Access Rights list box by selecting the set and clicking the less than sign (<) button.

You can remove a group from the Selected Groups list box by selecting the group and clicking the less than sign (<) button.



Only superuser or fullaccessusers can view or modify other user profiles. All lower access users can only view and modify their own profile.

Administrative functions can only be performed by fullaccessusers, including superuser. Admin functions include making changes to configuration (adding attributes, creating rules, etc.) and modifying commerce processes.

A Restricted Admin User will have no access to other areas of the administration platform: If a full access user is restricted from any product family or data table (by folder), the user will automatically lose access to other areas of the Administration Platform. He will only have access to the Product Family and/or data table that the superuser has allowed him access to.



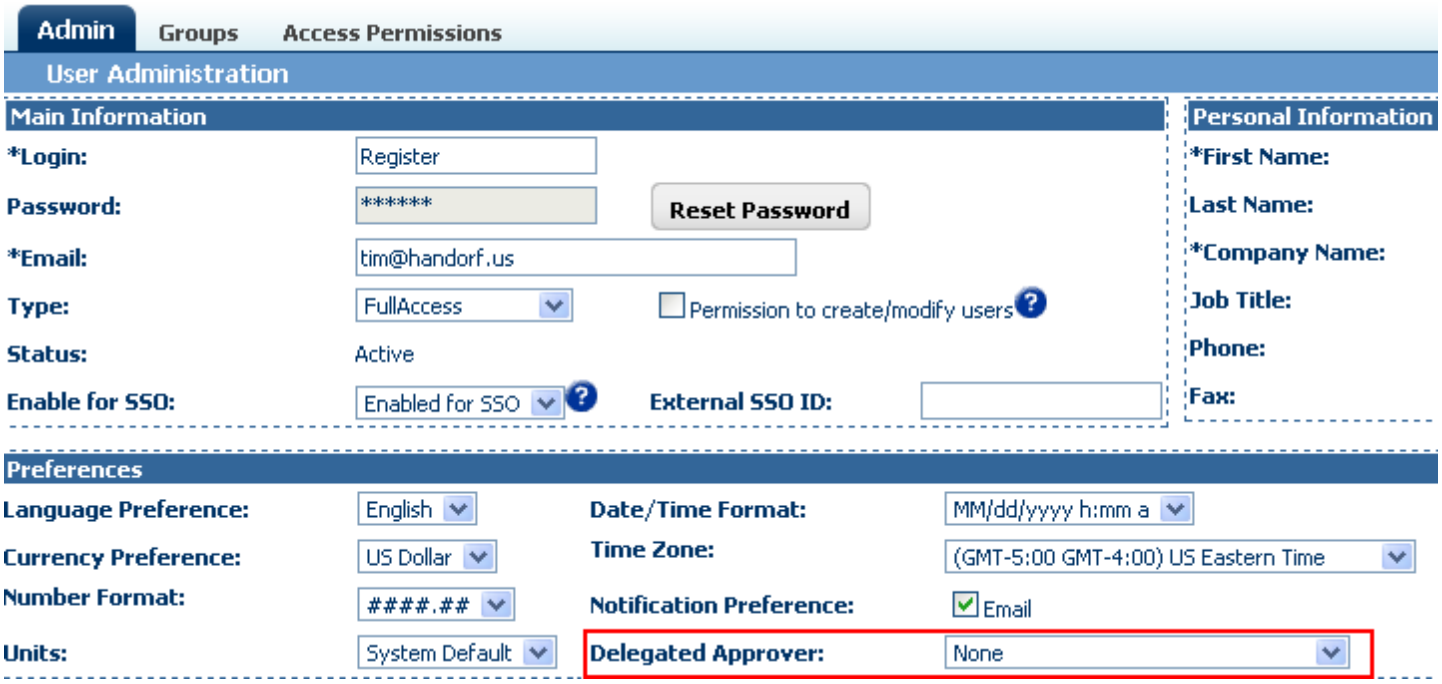
If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit. *When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.*

RELATED TOPICS

DELEGATED APPROVERS

OVERVIEW

Delegated approvers are sent notifications and allowed to approve/reject quotes if the user is an approver. If the user is approver by virtue of being in a group, delegated approvers cannot approve/reject quotes. In the user profile, you'll see a new attribute called Delegated Approver.



NOTES

If a user sets a delegated approver while they have quotes in their approval queue, the delegated approver will be able to approve/reject those quotes. They will not receive an email notification as they were not the delegate when the reason became active.

Delegated approvers will not be able to approve quotes if they do not have access to the transaction. This is because the delegated approver does not have access to a particular step.

Delegated approvers do not work for group approvals.

RELATED TOPICS

E COMMERCE REGISTRATION

OVERVIEW

eCommerce registration allows users and guests to configure and price products without having to create an account and log in. Upon navigating to the application home page, users will be able to begin selecting parts and/or configuring products without logging in, however, they are required to either log-in or create a user ID to check out. The default user will be: guest1@GuestCompany.

Another new feature is the Fully Customizable Quick Registration and My Profile pages. The fullaccessuser can upload XSL for customizing these pages.

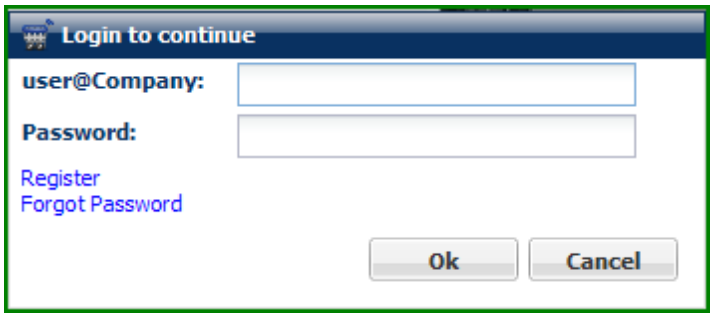
Click here for a demo of eCommerce Registration.

User-Side Functionality

Users would often like to configure an item and add it to the shopping cart when they are not logged in. Once the user has configured their product or has finished shopping, they will need to check-out. Upon check-out, users will be required to log-in to use previously saved information or they will need to register for a user account.



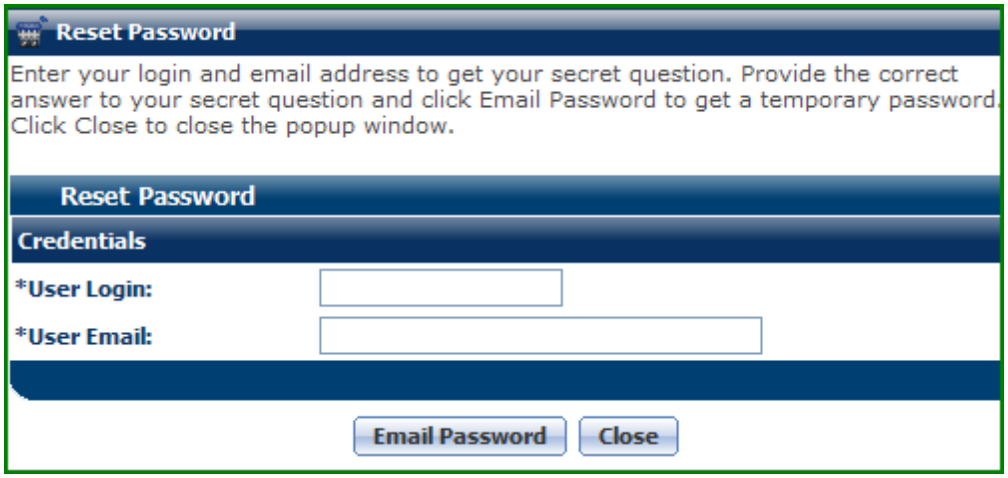
Upon clicking Check Out button (or any commerce invocation button), "not logged in" users will see this pop-up:



Registered Users

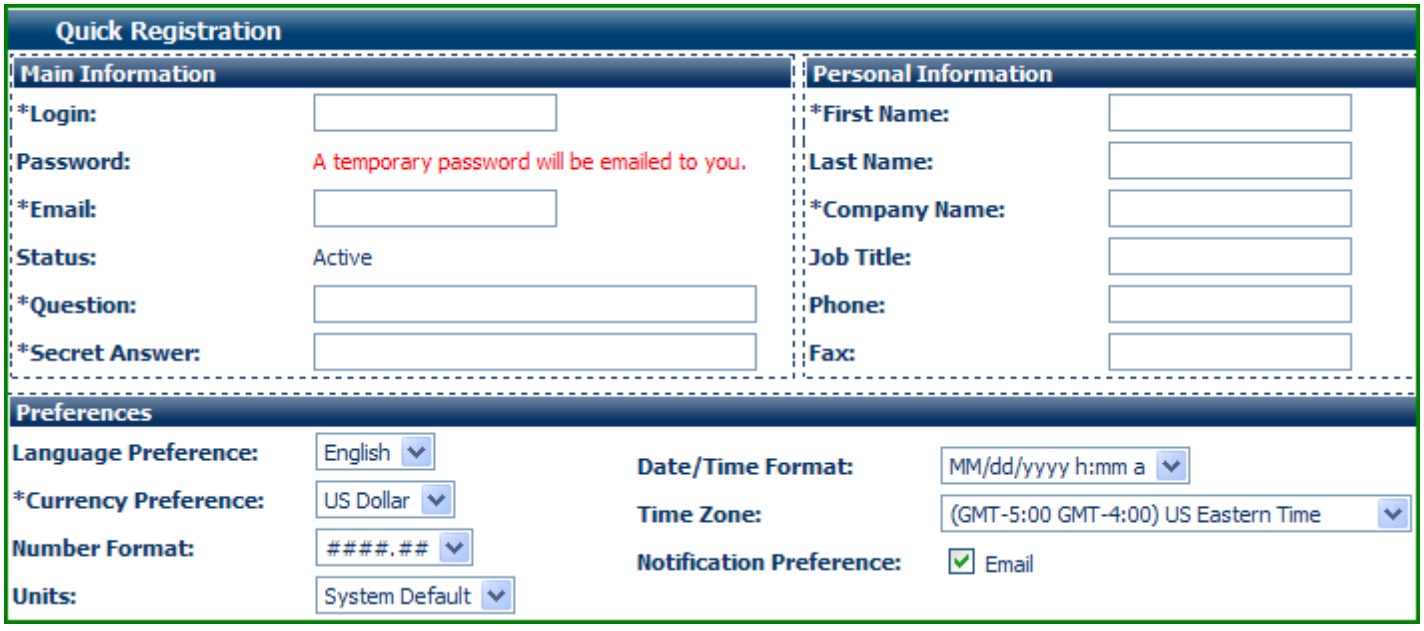
If the user is already registered, they will simply have to use their login credentials and the appropriate commerce invocation will take place. All items in the shopping cart will be added to the process. After the commerce process is invoked, all navigation buttons should appear that are assigned to the user type that logged in. Should a user try to login with invalid credentials, an error message will be displayed.

If the user cannot remember their password, they can click on the "Forgot Password" link. The following pop-up will be displayed, allowing them to have a temporary password emailed to them. Note: Users will be required to enter their user login as well as their user email address.



Guests

If the user is not already registered, they will need to click on the Register link. This will take them to a Quick Registration page where they will need to complete at least the required fields (login, email, question secret answer, first name, company name and currency preference). Once they are registered, the user should be logged in and all items will be moved to the appropriate commerce process.



ADMINISTRATION

Enabling eCommerce Registration for your Application

1. Navigate from the Admin Home Page > General > General Site Options.
2. Scroll to the section: "Login Options".
3. In order to enable guest access and registrations, you will need to select "Yes" for both Allow Guest Access and Enable Quick Registration.



USE CASES

Enabling users to add items to Shopping Cart without logging in

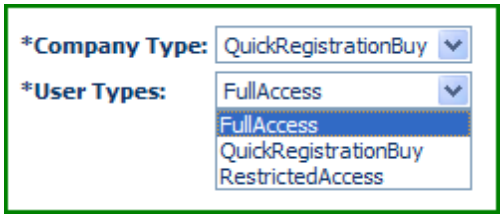
Users would often like to configure an item and add it to the shopping cart when they are not logged in. Once the user has configured their product or has finished shopping, they will need to check-out. Upon check-out, users will be required to log-in to use previously saved information or they will need to register for a user account. This use case assumes that Allow Guest Access and Enable Quick Registration have been set to "Yes" in the Global System Settings section described above.

1. Navigate from the Admin Home Page > Commerce and Documents > Process Invocation.
2. Select a **Page Title** of the page you would like to allow guests access to. For this example, we'll use "Model Configuration".
3. Select User Access Rights from the Navigation menu next to Add To Cart, and click List.



Verify that Add To Cart is in the Action list.

4. You will now see a new Company type and three new associated User types:



5. Choose **RestrictedAccess** from the User and Company type drop-down menus. This will allow users to add items to a shopping cart without logging in.



QuickRegistrationBuy as both Company and User type will allow users that registered on the Quick Registration page to create a transaction upon checkout.

NOTES



The admin user can rename existing fields on the Quick Registration and My Profile pages, but they cannot add new fields. Also, if the admin chooses to customize the Quick Registration page, they must also customize the My Profile page.

Periods can be used in BigMachines usernames.



The currency preference of the not logged in guest user is assumed to be the base currency of the applications. If the user changes their currency upon registration or login, then they will not be able to buy the items that have been placed in the cart. Users are only allowed to check out with items that have been configured with the same currency as that of the user during checkout.

RELATED TOPICS

CONFIGURATION OVERVIEW

OVERVIEW

Configuration is defined on the catalog hierarchy in BigMachines. For more information on the catalog hierarchy, see Catalog -An Overview. Catalog items contain regular attributes that cannot be modified at run time. Configuration enables the buyer to customize these items by specifying the values of configurable attributes. Configurable attributes can be created at all levels of the catalog hierarchy.

ADMINISTRATION

Situations appropriate for Rules

- **Guided Selection:** This is the process of guiding a user through a series of questions to help them search for one or more product line(s) within a product family or search for one or more model(s) within product line(s). This is akin to selecting a notebook model from a list of available brands.
- **Configuration:** This is the process of allowing users to customize a model to suit their needs. This is much like configuring the base notebook further by adding extra RAM, more hard disk space, warranty services etc. Some options may increase cost (pricing rules) and others options may not be possible (constraints) like getting a 3 month warranty with your model.

Types of Rules

The user experience in configuration is controlled by the ability to fire the appropriate rules based on user input. Different types of rules enhance the selection and configuration experience:

- **Recommendations:** System can inform the user about choices that you recommend based on their configuration.
- **Constraints:** System will discourage the user from making certain choices.
- **Recommended Items:** Up-sell products to the user, thus improving sales revenue.
- **Pricing:** Specify rules based pricing that helps you pass on area discounts, customization costs and other overhead to the customer.
- **Flows:** Layout attributes on the configuration pages so users are able to easily navigate and configure the selected models.

Best Practices

- Avoid using case sensitive text for displayed information or rule conditions (e.g. part numbers).
- Use Auto Updates sparingly.
- Avoid the use of JavaScript.
- Combine Config Rules when possible; separate Rules are more difficult to maintain and debug.
- Keep Rules on the same level, so they are easier to maintain and debug.
- Add relevant comments to explain specific design decisions.

- Avoid using custom templates.
- Create folders to organize your Data Tables.
- Avoid overriding menu attributes at lower levels in the product hierarchy.

NOTES



Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""



User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

RELATED TOPICS

CONFIGURATION PAGES - AN OVERVIEW

OVERVIEW

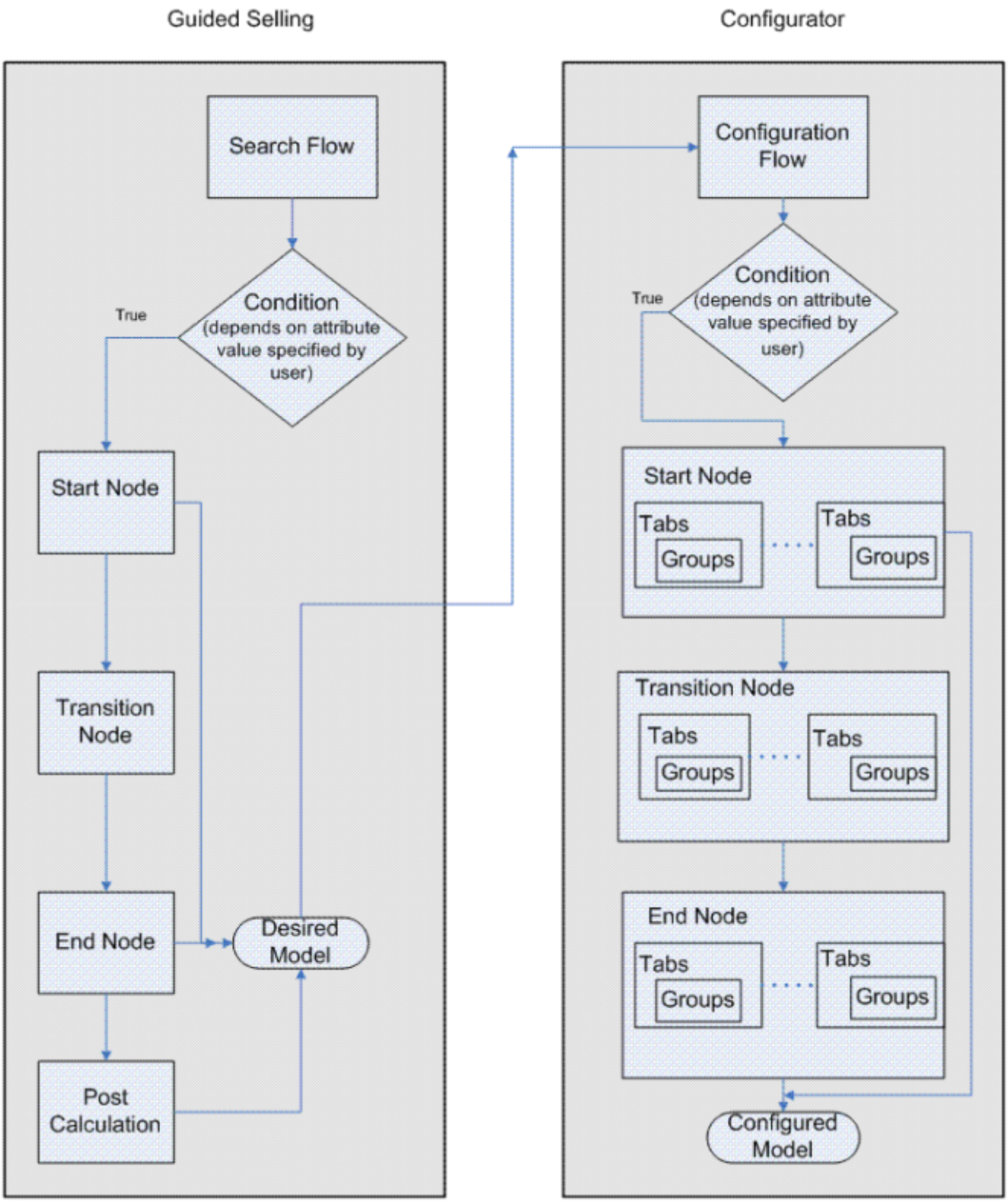
Flows are special rules that help guide buyers through the multi-step search and configuration process. *There are 2 types of flows:*

1. Configuration Flows
2. Search Flows

Configuration flows walk buyers through the product configuration process, enabling you to display configurable attributes in a logical sequence rather than all at once. A simple flow may contain a short process with only a single configuration step, whereas an advanced flow may contain a complex configuration process with many steps.

Search Flows, or selectors, enable you to create advanced search mechanisms for buyers. Within each segment, you can create multiple types of selectors to help buyers locate product lines, models across multiple product lines, or models within a single product line. A search flow gathers attribute values from the buyer and then uses those values to help buyers select suitable products. Search Flows are used for Guided Selling that helps target specific products with the desired configuration to the right buyer.

Page Workflow


Guided Selling
Search Flow
Condition
(depends on attribute value specified by user)
True
Start Node
Transition Node
End Node
Post Calculation
Desired Model
Configurator
Configuration Flow
Condition
(depends on attribute value specified by user)
True
Start Node
Tabs
Groups
Tabs
Groups
Transition Node
Tabs
Groups
Tabs
Groups
End Node
Tabs
Groups
Tabs
Groups
Configured Model

NOTES



If your system has configuration cache enabled, then all changes to flows data must be followed by a segment deploy in order to update the configuration cache.

RELATED TOPICS

CONFIGURATION TROUBLESHOOTING

OVERVIEW

My Configuration isn't working! This is a common situation that can cover many different areas.

> **Question:** Why isn't my change to auto-update reflected? *I have changed the "auto update" field on the attribute itself, but it is not reflecting the change.*
>
> **Answer:** If your attribute is mapped to a configuration flow layout, you have to go into that configuration flow and change the "auto update" value there as well. The change will not take affect until the value is changed in both locations.

General Config Troubleshooting

- Ensure that the Config cache has been deployed.
- Use the Pipeline Viewer to view attribute values and rules being fired.
- Check Rules using the Debugger. *Click a Favorite to locate the Debugger Import ID within the URL.*
- Is the Recommended Item rule firing? Check the Pipeline Viewer to confirm.
    - Change the logic on the Recommended Item rule to “Always True”. *If this fixes it, then revisit the logic that fires on the rule.*
    - Are all condition and input attributes active and present on the Flow? Check the advanced function on the Recommended Items action. Any attributes needed here must be on the Flow. *They can be hidden, but not inactive.*
- Is the Recommended Item rule returning a list of parts in the correct format?
    - Check the advanced function on the Recommended Items rule. Run the Debugger to check the return values.

ADMINISTRATION

Configurator recommending the wrong number of parts

- Is the Debugger returning the “correct” string, but the site is recommending the wrong quantity?
    - *Ensure that the advanced function's return string is formatted perfectly. Don't leave a |^| or anything hanging off the end of the string.*

Rule isn't firing

- Is the Product Family deployed?
- Are all the Condition Attributes present in the Flow?

- Is the rule active? Look at the Overall Status on the Rule List page. **It can be inactive for 2 reasons:**
    - *The Rule itself is marked as inactive.*
    - *One of the input Rules is marked as inactive.*



The status of the Rule is set on the Rule List page.

Change to Attribute's Auto-Update field not being reflected

If your attribute is mapped to a configuration flow layout, you have to go into that configuration flow and change the "auto update" value there as well. The change will not take affect until the value is changed in both locations.

Attribute change is not being reflected in the Config Flow

- Is Config cache enabled?
- Did you deploy?
    - *Ensure that the deploy button is shown on Config Quick Links for the selected Product Family.*
    - *Go to the Product Family Deployment Center to view the latest deployment time/date.*
- Is the Attribute being overwritten at a lower level?
    - *Check for a "Rollback" button on the lower levels of the Attribute detail page. If the button is present, the Attribute was overwritten at this level and is no longer inheriting changes from higher levels. The Rollback button will reset the attribute to its definition.*
- Is the value you added constrained?
    - *Check for related rules against the Attribute.*

Attribute value is different than expected

- Is the Product Family deployed?
- Is more than one rule acting on the attribute?
    - *Either alter Conditions so only one rule fires, or combine Rules. In general, simultaneously firing multiple Recommendation Rules causes confusion. Factors that determine behavior: Attribute order, Rules order, Rule levels, and number of Rule inputs.*
- Is the Data Table in your Rule deployed (if applicable)?
    - *Ensure that the latest modifications have been deployed if you're using a Table to locate the correct part number.*

Multiple updates required to change a value

- Has the order in which Rules fire and Attributes set been reviewed?
    - *The order of the attributes drives the order in which they are set. The order of the Rules order and number of condition Attributes determines the order in which they fire.*

Flow is different than expected

- Is the Product Family deployed?
- Where is the Flow located in the hierarchy?
    - *The lowest (most specific) Flow in the hierarchy should take precedence.*
- Are multiple Flow conditions True?
    - *The Flow with most condition inputs (most specific) should take precedence.*



To avoid confusion, make only one Flow "True" within a given situation.

Set Type is different than expected

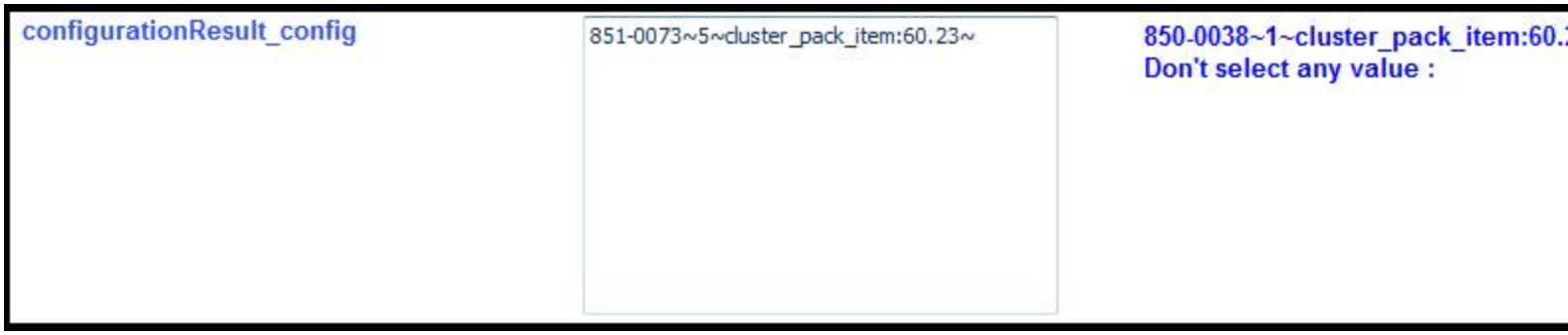
- If the set type on the attribute is set to "None", then the user's input will override the rule's value. *This causes the Rule to appear in the Pipeline Viewer, but not change the Configurator value.*
    - **Places to check the Set Type on the admin side:**
        - Attribute page
        - Rule page
        - Advanced Templates



If you have a custom template, ensure that you haven't forgotten about an old value remaining in the template. *The highest set type takes precedence for an Attribute. No matter where these are used: Force Set > Set > None*

Recommendation message unexpectedly appearing for text field containing delimiters

If a Recommendation Rule associated with an Attribute has a text datatype and the Attribute's recommended value BML script returns as a String, the system will display the tokens that are split along this delimiter sequence in the System Recommended message when the following character sequence is used: |^|



Using another character sequence will resolve the issue.



NOTES

User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

RELATED TOPICS

DEFINING PRODUCT SORT BEHAVIOR

OVERVIEW

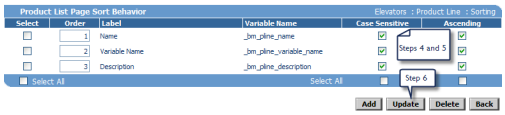
By default, items appear on the product line and model search results in order based on their internal system ID. You can change this by designating attributes you prefer the system to sort products by. To display products in as random an order as possible, create a sort order with attributes whose values fall in a narrow range.

ADMINISTRATION

1. Navigate from: Product Line Attribute List page > List Ordering.
2. Click **Add** to open the Add Product Attributes for Sorting page.
3. Select one or more attributes in the **Unsorted Attributes menu** and click **Add**.



4. Select the **Case Sensitive** checkbox corresponding to text attribute types.
5. Select the checkbox under the **Ascending** column.
6. Click Update to save changes to the page and click Back to return to the Product Attribute List page.



NOTES

To stop using a sort attribute, Select the checkboxes corresponding to the attributes you no longer want to use for product sorting and click Delete.

RELATED TOPICS

SETTING UP PRODUCT FAMILIES

OVERVIEW

Products in BigMachines are structured in a hierarchical fashion. The first step in Product Definition is to create Product Families. Product families are broad classifications of products, typically based on industry standards, rather than a single company standard. Product family classifications could take into consideration how you want to merchandise your products, or how you want users to navigate through the catalog. The BigMachines system supports up to 30 product families.

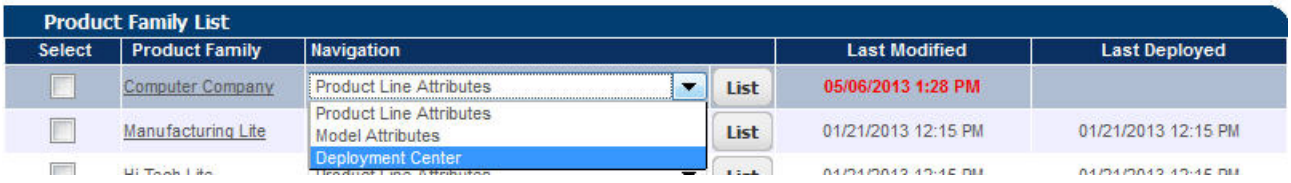
ADMINISTRATION

Step 1: Creating a Product Family

1. Navigate from: Admin Home Page > Products > Product Definition.
2. Enter the **name** of the product family in the Label text box.
3. Enter a **Variable Name**. (Must be alpha-numeric and no spaces or special characters are allowed).
4. Click **Add**.

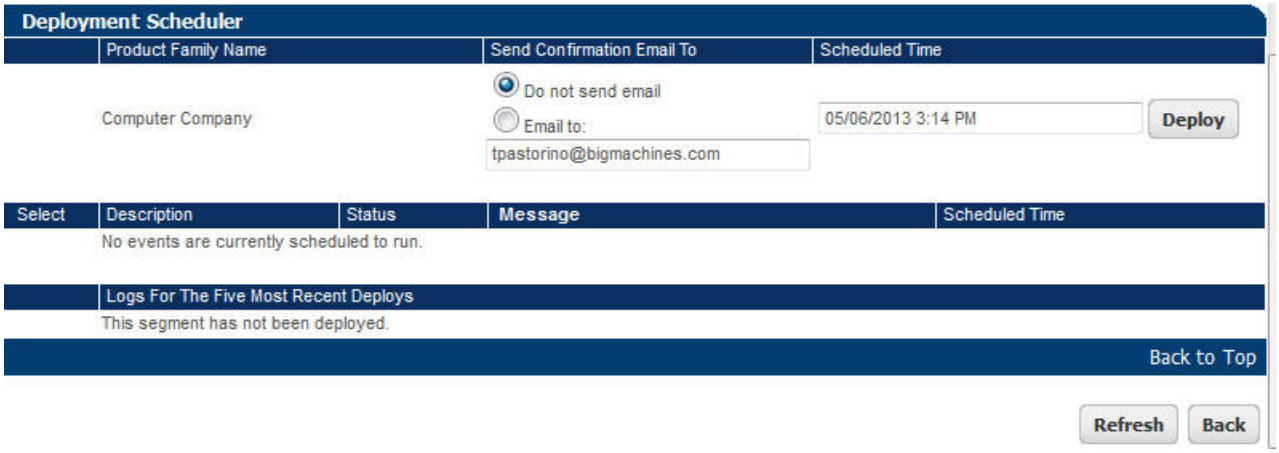


5. This will bring you back to the Product Family List page. You will now need to deploy your Product Family. To do so:
6. Select the **product family**.
7. Choose **Deployment Center** from the drop-down menu in the Navigation column.
8. Click **List**.



9. Decide if it will be necessary to send out confirmation emails every time a Product Family is deployed. If so, make sure to select the "Email to" radio button.
10. Enter a **valid email address** in the text box.

11. Click **Deploy**. *You can schedule a time for the future if in the text box next to the Deploy button.*

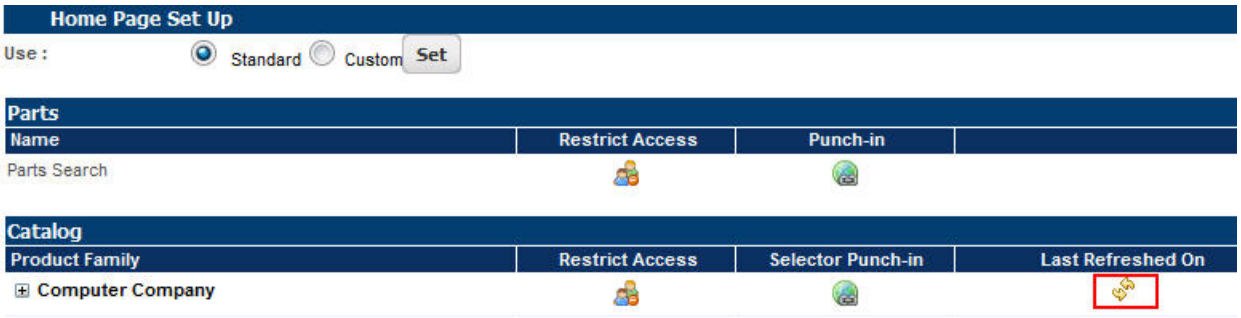


Step 2: Activating the Product Family

1. Navigate from: Catalog Definition: Admin > Configuration > Catalog Definition
2. Select **Product Families** and click **List**.
3. Click **Add**.
4. Choose your **product family(ies)** from the menu and click **Add**.
5. Find your newly Supported Product Family and select Deployment Center from the drop-down Menu.
6. Decide if it will be necessary to send out confirmation email. If so, make sure to select the "Email to" radio button.
7. Enter a **valid email address** in the text box.
8. Click **Deploy**. *You can schedule a time for the future if in the text box next to the Deploy button.*

Step 3: Deploy the Home Page

1. Navigate from: Admin > Application Branding and Home Page > Home Page.
2. Click the **Refresh icon** next to your product family.



3. Click on **Deployment Center**.
4. Decide if it will be necessary to send out confirmation email. If so, make sure to select the 'Email to:' radio button.
5. Enter a **valid email address** in the text box.
6. Click **Deploy**. *You can schedule a time for the future in the next box next to the Deploy button*.



You should now be able to view your new Product Family in your Home Page.

Step 4: Adding or Editing a Product Line

1. Navigate from: Admin Home Page > Products > Catalog Definition > All Supported Product Families > List > Select a Product Family > List > Product Line > List.
2. Click **Add**.
3. Enter a **unique product line** name .
4. Enter a **unique variable name** (when adding product lines ONLY).
5. Enter a description OR click Edit HTML to open the Content HTML Editor. Here you can customize the description, add images and hyperlinks and much more.

Content HTML Editor

The Content HTML Editor is similar to a Rich Text Editor. In the Design Mode, you can perform standard functions: save, search, print, find and replace, spell-check, cut and paste, redo, undo, etc. You will also be able to add and modify text. If you wish to modify the content with HTML, simply click on Source Mode and begin writing your HTML script.

Click the Save button the bottom of the page to Save your changes and close the Editor. Click Close to close the Editor without saving changes.

Step 5: Adding or Editing Models

From the Product Line Administration List, select a Product Line for which you would like to add or edit Models.

Existing Models

1. Select Model from drop-down navigation menu.
2. Click **List**.

New Model

1. Click **Add**.

1. Enter a unique model name.
2. Enter a unique variable name (when adding models ONLY).
3. Enter a description **OR** click **Edit HTML** to open the Content HTML Editor.

USE CASES

Removing Support from a Product Family

1. Navigate from: Admin Home Page > Catalog Definition > All Product Families > List > Supported Product Families List Page.
2. Select the product family(ies) from which you'd like to remove support by clicking the checkbox next to their name.
3. Click **Remove Support**.



If you get an error message that reads: "Could not remove support for [product family]. Please delete the backup first.",click here and follow these steps.

1. Navigate to Admin Home Page > Catalog Definition > All Product Families > Deployment Center > List.
2. Click the **Delete** button under the Backups heading.



Only do this if absolutely necessary. All back-ups will be removed.

RELATED TOPICS

PRODUCT FAMILY HIERARCHY

OVERVIEW

Products in BigMachines are structured in a hierarchical fashion. The first step in Product Definition is to create Product Families. Product families are broad classifications of products, typically based on industry standards, rather than a single company standard. Product family classifications could take into consideration how you want to merchandise your products, or how you want users to navigate through the catalog.

All of your product lines could belong to a single, all-encompassing product family, or they could be split up into several product family. For example, you can create general product families, like ”pumps”, or you can create more specific product families, like ”centrifugal pumps” and ”steam pumps”. Depending on the product families created, some catalog page templates and home page flow templates may be more appropriate to use than others. For more information on template choices.

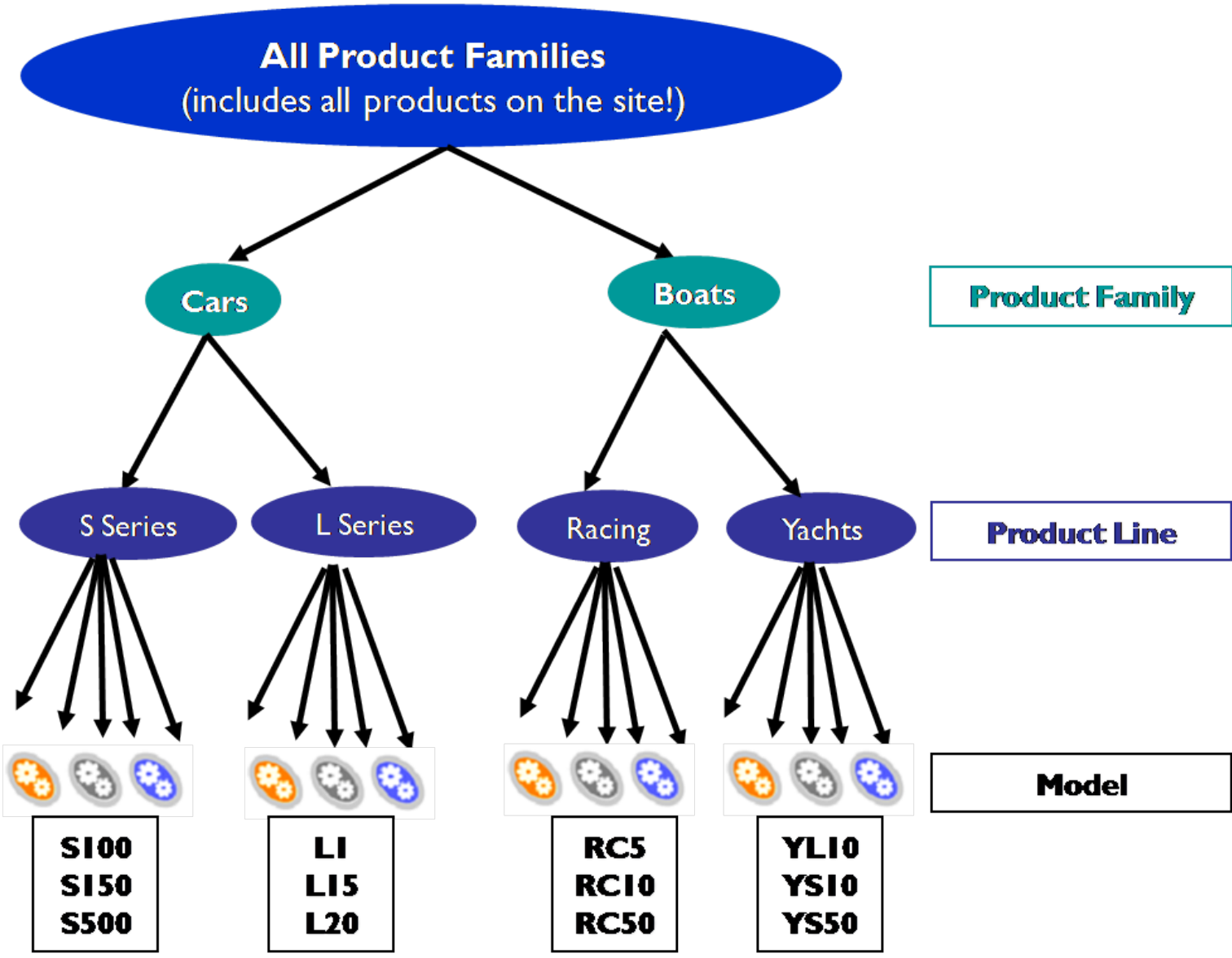
Product Line and Model Attributes capture the general characteristics of the product. By default, product families and product line/-model attributes are created in an inactive, pending state. This enables you to create product family over time, and then deploy them when they are complete. Deployed product families appear on the commerce-side and in Supported Products administration, where product line and model data can be uploaded.

Product Family Hierarchy

There are four tiers in the products hierarchy. You can define attributes, rules and flows at any level in the hierarchy and these will be inherited by the product's direct descendants.

Level	Description
All Product Families	This is the highest level in the products hierarchy. Configurable attributes, rules and flows created at this level are inherited by all products below.
Product Family	Individual product families inherit the configurable attributes and configuration rules from the All Product Families level. Anything defined at this level is accessible by Product Lines and Models within this Family
Product Line	Attributes, rules and flows are inherited from All Product Families and parent Product Families. Anything defined at this level is accessible by Models within the Product Line.
Model	Individual models inherit attributes, rules, and flows from all of the above levels.This is the lowest level in the products hierarchy, which means anything defined at this level is NOT inherited anywhere.

Product Hierarchy Example



NOTES

Best practice is to NOT define anything at the MODEL level unless absolutely necessary. This is to reduce the maintenance required to update attributes/rules/flows in multiple places each time.

In fact, some experts think that the best way to go is to define only one Product Line and Model per Product family and use configurable attributes to store the Product Line and Model name instead. This



approach will reduce the number of rules that you need to maintain, but one downside is that those few rules can be quite a bit more complicated than if you were to explode your products into the standard hierarchy Choose wisely.

Typically, you will want to define attributes and rules at the HIGHEST level possible.The Product Family level is the most ideal.



There are no limits to the number of product families, lines or models that can be created.

RELATED TOPICS

PRODUCT ATTRIBUTE GROUPS

OVERVIEW

Groups enable you to organize attributes. New product line/model attributes are always created in the default "General" group. When a product family contains a lot of base attributes, custom groups can be created to organize how they appear on both commerce and administration pages.

Product line/model attributes can be grouped in two short steps. The first step involves creating custom groups. The second step involves moving attributes from one group to another. You can also order how groups appear on a page.

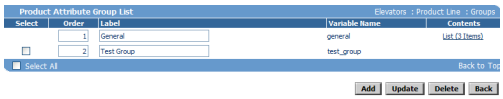
ADMINISTRATION

Adding a Product Attribute Group

1. Enter a name or **Label** for the product attribute group.
2. Enter a unique **Variable Name** for the group.
3. Click **Add**.



4. Click **Back** to return to the Product Attribute Group List page.
5. New groups will not contain any attributes, so you will need to add them. In the image below, you will notice that in the far right corner under Contents, the General group contains 3 items and the Test Group just created doesn't contain any items.





This will automatically refresh the page and you will notice that the selected attribute is no longer visible.

Adding an Attribute to a Product Attribute Group

1. Click the **List (x items)** under Contents of a particular group.
2. Use **Select/Select All** to select the attributes you wish to add to your new group.
3. Choose your **group** from the drop-down menu and click **Move**.
4. Click **Back**. *You will notice that the new group that didn't have any items in its contents now has a link to the attributes it contains.*

5. Click **List** to verify.



NOTES

Attribute moves take effect after the product family is deployed.After moving an attribute, the "moved from" group remains visible.

To verify that the attributes did move to the destination group, return to the Attribute Group List page and click the List button related to the destination group.

After creating groups, the parent product families must be deployed in order to active them.

Group order changes and name edits take effect after the product family is deployed from the Product Family List page.

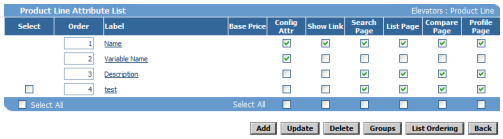
RELATED TOPICS

PRODUCT LINE AND MODEL ATTRIBUTES

OVERVIEW

Product line/model attributes characterize general product traits. For example, if the product family is an Automobile, then the product line/model attributes might be wheels and brake type.

The purpose of creating product line/model attributes is to provide an attribute framework for the product lines and models within each product family. Using this framework, product upload fullaccessusers can differentiate items by entering product-specific values into these common sets of attributes. Product line/model attribute sets also establish the attributes partner organization's users can use to conduct product searches within product families.



What can be accomplished using the Product Attribute List page

- Change the order of the attribute by modifying the Order field and clicking Update to refresh.
- Click a label to display the Attribute Editor page. See Creating and Maintaining Product Line/Model Attributes.
- Make attributes available for use in configuration rules by selecting the required checkbox for Configuration Attribute.
- Select the checkbox for Base Price to make an attribute the base price.
- Make the attribute visible on other pages by selecting the appropriate checkbox.
- Click Add to create a new attribute.
- Click Update to save changes to the page.
- Click Groups to see the Product Attribute Group List page for managing product attributes groups.
- Click List Ordering to see the Product List Page Sort Behavior page that can be used to define the sort order for products found during a search.
- Click Back to return to the Product Family List page.

ADMINISTRATION

Default Attributes

The system automatically generates a default set of product line/model attributes for each product family. It is not possible to delete default attributes.

Default Attributes automatically created for each Product Family

Attribute Name	Field Type	Description
Name	String	This field accepts product line/model names. The system validates it to make sure each name is unique. Names can be up to 255 characters.
Description	Long String	Product or Model descriptions entered into this field can exceed 255 characters.

Additional product line/model attributes can be added to these default sets. The commerce system supports up to 60 product line attributes and 60 model attributes for each product family.

Other types of Product Line & Model Attributes

Field Type	Description
Boolean	Boolean fields display as a set of radio buttons. This field is used to construct an either/or attribute.
Text Area	Long strings display as text boxes that exceed 255 characters.
Text Field	String displays as text fields that cannot exceed 255 characters.
Single-Select Menu	Single-select menu fields offer a selection of values from which you can choose a single item.
Multi-Select Menu	Multi-select menu fields offer a selection of values from which you can choose multiple items.
Integer	Integer fields validate that whole numbers are entered. The system can also validate within a range of integers.
Decimal Number (Float)	Float fields validate numbers with decimal point values. The system can also validate within a range of float values.
Range	Range fields enable you to set a lower and upper set (min/max) of values for a single attribute. During a product search, the system checks wither the user-submitted value falls between the lower and upper bounds of the range. If it does, the product is returned in the search set.
File Attachment	File attachment fields validate a file extension. The system can specify some particular file extensions.
Currency	Currency fields validate numbers with decimal values. The system can also validate currency as non-negative.

After generating product line/model attributes, you can create groups to organize how they appear on pages such as the Attribute Search page (in the commerce area) and the Product Line Administration page (in the Supported Products administration area). You can also order groups and attributes, and define an attribute sort order that instructs the system how search results should display at both product line and model levels. You can include a product line/model attribute as base price. You can also ensure that an attribute is available as input for configuration rules.



Similar to product families, product line/model attributes are also created in an inactive, pending state. They become active when the parent product family is deployed.

RELATED TOPICS

DEFINING PRODUCT SORT BEHAVIOR

OVERVIEW

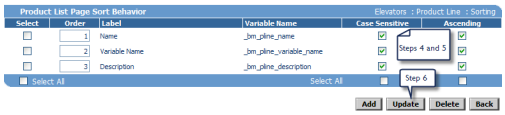
By default, items appear on the product line and model search results in order based on their internal system ID. You can change this by designating attributes you prefer the system to sort products by. To display products in as random an order as possible, create a sort order with attributes whose values fall in a narrow range.

ADMINISTRATION

1. Navigate from: Product Line Attribute List page > List Ordering.
2. Click **Add** to open the Add Product Attributes for Sorting page.
3. Select one or more attributes in the **Unsorted Attributes menu** and click **Add**.



4. Select the **Case Sensitive** checkbox corresponding to text attribute types.
5. Select the checkbox under the **Ascending** column.
6. Click Update to save changes to the page and click Back to return to the Product Attribute List page.



NOTES

To stop using a sort attribute, Select the checkboxes corresponding to the attributes you no longer want to use for product sorting and click Delete.

RELATED TOPICS

ARRAY BEST PRACTICES

OVERVIEW

Building Arrays

1. Configurable array sets must be created the proper way in order for the +/- add/remove buttons to appear or work .
    - Create the array set BEFORE visiting the “Set” tab of the array attribute
    - Ensure that the array control attribute is in the same group as the array set, and is not hidden.
    - This order will always work:
        a. Create the array control attribute
        b. Create the array set
        c. Create the array attributes and add to the set
2. If you created the wrong way you DO NOT need to delete your attribute and recreate. Here are the steps to fix an Array Set that is not showing the increment / decrement buttons:
    - Do the following for each array in the array set:
        a. Go to the Array Attribute on the admin side.
        b. Go to the Set Properties tab.
        c. Change the Array Set from the real one to the [DEFAULT] one.
        d. Change the Array Control Attribute to the blank entry.
        e. Click Apply. *important step*
        f. Change the Array Set back to the real one.
        g. Click Update.
    - Once you have done that for all arrays, the little icons should return.
3. It is possible to have one Size attribute control multiple arrays, but not recommended, since it will lead to confusion in maintenance.

Recommendation Rules and Arrays

- Set Type Recommendations will work for an array attribute only if none of the values of those attribute elements is changed by the user. Once changed, the system treats it as an overwrite of the attribute and will not perform a Set recommendation on any elements that get added afterward.

- If you recommend a value for a string array of "" (blank) or $BM_UNCHANGED_STR$ (leave value unchanged), even if the value is blank, you will still get the recommendation message ("Don't select any value…").

User Experience with Arrays

- There is quite a bit of BM behind-the-scenes Javascript associated with array sets. The presence of many array sets will affect performance, especially in IE.
- When you add a new element to an array using the +/- buttons, the value it takes is the value of the previous last element (the now penultimate element). Defaults are not respected in arrays.
- The recommendation text for an array attribute always shows the variable name of the attribute, rather than the display name, as for non-array attributes.
- Maximum number of rows in an array is 200.
- Multi-select menus are not possible in arrays.

Arrays & Table-based Rules

- Simple Table-based Rules operate per index; they will not operate across indexes.
- Attribute type limitations are consistent with existing simple Table-based Rules.
    - If the Table column is numeric, then the Array Attributes must be numeric

---

NOTES



Maximum Array size is 200.

RELATED TOPICS

CONFIGURATION ATTRIBUTES OVERVIEW

OVERVIEW

Configurable attributes are used to define those characteristics of product families that can be specified by the buyer. Much of the configuration system's functionality requires the use of these attributes. They are used in search flows, configuration flows and every type of configuration rules.

Configurable attributes can be created as menus or text fields. Menus enable you to designate specific values (a discrete set of values) from which a buyer can choose one or more items. Text fields are more open-ended allowing the buyer to type in values. Attributes can be of float, integer, date, HTML or text data type. You also can choose the status of the configurable attributes as active, inactive or internal. Configurable attributes can also be made active during certain time periods by specifying start and end date

Configurable attributes can be defined as stand-alone attributes or they may be defined as arrays. Arrays attributes help in grouping multiple scalar attributes. Usually, the size of an array attribute is specified by a control attribute.



What can be done on this page

1. Delete Attributes
    - Under the Select column, select the corresponding checkboxes for the attributes you want to delete and click Delete to remove the required attributes.

2. Re-order Attributes
    - Re-order the attributes by modifying the numbering in the Order column.
3. Add Attributes
    - Click Add to open the Attribute Editor page for adding a configurable attribute.
4. Edit Attributes
    - Click a Variable Name link to open the Attribute Editor page that displays the General tab by default for viewing and editing the current information. The other tabs available are Related Rules, Template Properties and Set Properties. The Set Properties tab appears only when the configurable attribute is of the type, Array.
5. Change Attribute Status
    - Modify the Status using the drop-down list provided. The options are: Active, Internal, and Inactive.
6. View Related Rules
    - Click List under the Related Rules column to open the Attribute Editor page that displays the Related Rules tab by default for viewing and editing the current information.
7. Manage Array Sets
    - Click List Array Sets to go to the Configurable Array Sets Administration List page. Click Update to save the changes made. Click Back to go back to the Supported Products page for configuring catalog definitions.

ADMINISTRATION

Configurable Attribute Calculators

Attribute calculators are used to help buyers determine attribute values. *After you create an attribute calculator, a calculator icon appears next to the attribute. When someone clicks the icon, the calculator launches.* The calculator contains question sets. The calculator uses the input entered into a question set to compute a single attribute value, and then auto-populate the attribute field with the value. Within each attribute calculator, you can create multiple question sets. *Depending on the shape of an item, i.e. square vs. conical, volume can be calculated differently. Multiple question sets can resolve such differences.*

Example: you can create an attribute calculator for a ”volume” attribute by creating a question set that contains length, width and height measurements. Using the three variables, the calculator can determine a volume value.

Creating a Calculator

1. Navigate to the **Configurable Attributes Calculator Administration** page.
2. Add configurable attributes.
3. Create a question set.

Adding a Calculator

1. Navigate from: Supported Product Families page > Configurable Attribute Calculator > List. *Configurable Attribute Calculator Administration List page is displayed.*
2. **Add** a configurable attribute calculator.
3. **Add** configurable attributes to the configurable attribute calculator. *The attribute you have created is displayed in the **Configurable Attribute Calculator Admin List**.*
4. Click **List** against the attribute you created. *This directs you to **Calculator Tabs** screen.*
5. Click **Add** to create a tab. ***Calculator Tab Editor** is displayed.*
6. Enter the **Title** and **Tab Name**. *Attach a file if required.*
7. Enter the **Header Text**.
8. **Add** the tab and click **Next**. ***Calculator Attribute Administration List** is displayed.*
9. **Add** the attribute. ***Select Field Type for Attribute** screen is displayed.*
10. Select the type of field from the displayed list and click **Next**. ***Calculator Attribute Administration** screen is displayed.*
11. Enter the **Name** of the attribute.
12. Specify the **Variable Name**.
13. Enter the appropriate **Description**.
14. Enable the **Required** field you wish the users to fill out on the buy-side.
15. Select the **Validation Type** from the drop-down menu.
16. Specify the **Maximum** and **Minimum** values.
17. Select the conversion group from the drop-down menu.
18. Select the units.
19. Click **Add**.
20. Click **Define Function** within the Calculator Attribute Administration List after defining the necessary attributes. *This directs you to the BML Editor.*
21. Define your function and click **Run** to compile the code.
22. Click **Save** to save the code. *This directs you to **Calculator Tabs** page. Click **Preview** to view how the calculator appears on the buy-side.*

Question Sets: A single calculator can be created for each attribute. Within each calculator, you can create multiple question sets. Each question set displays as a separate tab in the calculator. Navigate from a calculator administration page.

Viewing Question Sets

1. Choose the attribute question set you wish to view on the **Configurable Attributes Calculator Administration List** page.
2. Click **List**.
3. Click a question set name to view the question set. *The **Calculator Tabs** page shows the names of the question sets created for the attribute calculator.*
4. Click **Next** on the **Attribute Calculator Editor** page.
5. Click **Next** if you wish to view the function created for the input requested from the question set. The Questions List page shows the questions created for the question set.



You can view each question set and each function created for a question set individually.

Editing Question Sets

1. Choose the attribute question sets you wish to edit on the **Configurable Attributes Calculator Administration List** page.
2. Click the question set name of the question set you wish to edit. *The **Calculator Tabs** page show the names of the question sets created for the attribute calculator.*
3. Do one of the following:
   - *To edit the tab that displays the question set:* Edit any of the information displayed in the **Attribute Calculator Editor**, and click **Save**.
   - *To edit individual questions in the question set:* Click the question name you wish to edit.
   - *To edit the function created for a question set:* Click **Next**. Edit the function created for the question set in the **Function Editor**.



The question sets within an attribute calculator can be edited at any time.

Deploying Question Sets

Question sets must be deployed to become active in other areas of the commerce system. Navigate from a calculator administration page.

1. Choose the attribute question sets you wish to deploy on the **Configurable Attributes Calculator Administration List** page.
2. Click **List**.
3. Choose the question sets you want to be visible in Commerce, and select the corresponding checkboxes under the **Show to Buyer** column to deploy a question set. *The **Calculator** page show the names of the question sets created for the attribute calculator.*
4. Click **Update**.



You can set visibility guidelines for attribute calculators.

Deleting Question Sets

You can delete both entire question sets and individual questions within question sets.

1. Choose the attribute question sets you wish to delete on the **Configurable Attributes Calculator Administration List** page.
2. Click **List**.
3. Do one of the following:
   - *To delete a question set:* On the Calculator page, choose the question set you want to delete, and mark the check box under the Delete column. Then click Delete.
   - *To delete individual questions:* Click the question set whose questions you want to delete. Click **Next**. Select the corresponding checkboxes you wish to delete found under the **Delete** column on the **Parameter Assistant Questions List** page. Then click **Delete**.



Entire attribute calculators can be deleted by removing the attribute name from the calculator administration page.



Use caution when deleting question sets because it is not possible to recover deleted data.

If the attribute for which a calculator has been created changes from string to numeric, the attribute calculator is automatically deleted.

Reordering Configurable Attributes

1. Navigate from: Admin > Catalog Definition > All Product Families (List) > desired Product Family
2. Select **Configurable Attribute** from the drop-down menu.
3. Click **List**.
4. Enter the desired order number of each Configurable Attribute in the **Order** field.



Configurable Attributes Administration List — Product Family : Comput

Select	Order	Name (Variable Name)	Array Type	Pricing Type	Attribute Type	Data Type	Set Type	Level	Status	Overall Status	Rel Rul
		CRM Account Id (crmAccountId)	No	No	Text Field	Text	None	*All Products	Active	Active	List
		Run Once Begin (runOnceBegin)	No	No	Boolean	Boolean	None	*All Products	Active	Active	List
		Run Once End (runOnceEnd)	No	No	Boolean	Boolean	None	*All Products	Active	Active	List
☐	1	Text Test Attribute (textTestAttribute)	No	No	Text Field	Text	None	Product Family	Active	Active	List
☐	2	Integer Test Attribute (integerTestAttribute)	No	No	Text Field	Integer	None	Product Family	Active	Active	List
☐	3	Float Test Attribute (floatTestAttribute)	No	No	Text Field	Float	None	Product Family	Active	Active	List
☐	4	SSM Test Attribute (sSMTestAttribute)	No	No	Single Select Menu	Text	None	Product Family	Active	Active	List
☐	5	MSM Test Attribute (mSMTestAttribute)	No	Yes	Multi Select Menu	Text	None	Product Family	Active	Active	List
☐	6	HTML Test Attribute (hTMLTestAttribute)	No	No	Read Only	HTML	Forced Set	Product Family	Active	Active	List
☐	7	Currency Attribute Test (currencyAttributeTest)	No	No	Text Field	Currency	None	Product Family	Active	Active	List
☐	8	Date Attribute TEst (dateAttributeTEst)	No	No	Date	Date	Forced Set	Product Family	Active	Active	List

Translations | Export | Add | Update | Delete | List Array Se

5. Click **Update**.

Order of Operations: Rule Precedence

1. Config Flow Rules
2. Constraint Rules
3. Recommendation Rules
4. Constraint Rules (again)
5. Hiding Rules
6. Recommended Item Rules
7. BOM Rules

Attribute Order & Rule Precedence

1. The attribute order affects rule order in the following ways:
2. Config rules with "True" conditions are reviewed.
3. The rules that reference the first Configurable Attribute are run.
4. The rules that reference the second Configurable Attributes are run. *Rules on subsequent attributes are run.*



Order becomes important when there are dependent Recommendation Rules that need to be calculated in a specific sequence. Example: D is the sum of B and C; C = A*2. If C appears after D within the Attribute List, D's calculation will use a stale value for C, resulting in an incorrect value for D.



Rules are executed in the order that their related Configurable Attribute appear within in the Configurable Attributes Administration List. *In general, most users will not need to reorder their attributes; only a few special circumstances warrant attribute reordering.*

NOTES



Avoid overriding menu attributes at lower levels in the product hierarchy.

Have as few attributes "auto-update" as possible.



Attribute range calculations are validated on the Value Punch-In. **Example:** if the value was passed in the URL to the Configuration, it will be validated and an error message will appear. *This affects both Mobile and Desktop.*

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""



Don't have displayed information be case sensitive or be a condition for a rule (i.e., Part Numbers).

RELATED TOPICS

MANAGING ARRAYS AND SETS

OVERVIEW

Array attributes are used to easily group multiple, scalar attributes using a single data structure, and are available both for the selection and configuration process. Similar to other configurable attributes, you can create array attributes at all levels in the configuration hierarchy, but can delete them only at the level of the hierarchy in which they were created. You can create array attributes for all the available data types . When default value is set for an array attribute, the value is applied to all elements of the array. It is not possible to set default values for attribute elements at the attribute definition level. When you want to set different default values for the various elements of an array, you must define a recommendation rule.

When you create an array attribute, it automatically gets associated to a default set, which can then later be changed to a custom set. Array attributes are placed in Sets for easier and more intuitive creation of configuration flows. This is optional and if you do not associate the set with a size, the system assumes that the size is always 1 on the user-side.

Characteristics of Array Attributes & Sets

1. You can associate array attributes to only those sets that have been created at the same level.
2. You can associate one attribute to multiple sets.
3. You can modify associate of sets after the initial creation of the attribute.
4. You can view the associated attributes for the set and a listing of the unassociated attribute, in the Configurable Array Sets Editor page. If you remove all the associated attributes from a set, the complete set is hidden in the configuration flow.

While defining array attributes, you must use the **Set Size attribute** to define the size of the array. A Set Size attribute is a special kind of an integer attribute that has been marked as an Array Control Attribute. If you do not select the set size attribute, the system throws a warning that the array created will always be of size ”1” until a size attribute is associated.

While creating array control attributes and array attributes, make sure that they are ordered such that the control attribute has a lower order number than the array attribute it drives. This ensures that the rules fired have the desired effect (as always fires rules based on the order of the attributes).

Characteristics of Sets association with the Size Attribute

1. If you associate an array attribute to a set after having already selected a Set size attribute, the associated set’s size attribute is shown under the General Tab of the Array Attribute Editor.
2. You can create multiple sets at the same level and associate them to a single set size attribute.
3. The Set Size Attribute fields (in the Set Properties Tab, or the Configurable Array Set Editor page) displays only those integer attributes that have been marked as Array Control Attributes and have a range validation from 1 to 50.

Array Control Attribute

An array control attribute is used as the Set Size Attribute. They are used to define the size of the array. An array control attribute is a special kind of integer attribute that has been marked as an Array Control Attribute. When you create an attribute with an integer data type and a text field attribute type, there will be a new field called Array Control Attribute. Checking this box marks the attribute as available to use as a Set Size Attribute for use in Array Sets.

While creating array control attribute and array attributes, make sure that they are ordered such that the control attribute has a lower order number than the array attribute(s) it drives.



One control attribute can be associated to multiple sets

ADMINISTRATION

Setting Up Array Sets

1. Click on **List Array Sets.**
2. Click .**Add**
3. Enter a **Name**.
4. Select the appropriate **size attribute**. (Remember: The Array Control Attribute is used to determine the Size Attribute).
5. Choose the **Display Type**: Horizontal, Vertical or Tabbed (See Set Properties Tab for more details).
6. Select the display of labels using the options in the **Label Option** menu:
    1. Display only First Label
    2. Display Both Labels and Indices (only allowed with the Tabbed display type)
    3. Display only Labels (only allowed with the Tabbed display type).
7. Define a postfix and/or prefix for each index shown within in each set. *This allows you to create a group label for each iteration of the array attributes within the set.*
8. In the Set Association Information section, a list of unassociated and associated attributes is displayed. You can move the attributes from the unassociated to the associated section, or vice-versa, using the right and left arrow buttons.
9. Click **Apply** to save the changes and remain on the current page or click Update to save the changes and return to the Attributes List page.



There is a tab at the bottom of the editor labeled Translations. In this tab, you can manually enter text translations for supported languages. You can only set the Name, Description, Prefix and Suffix.

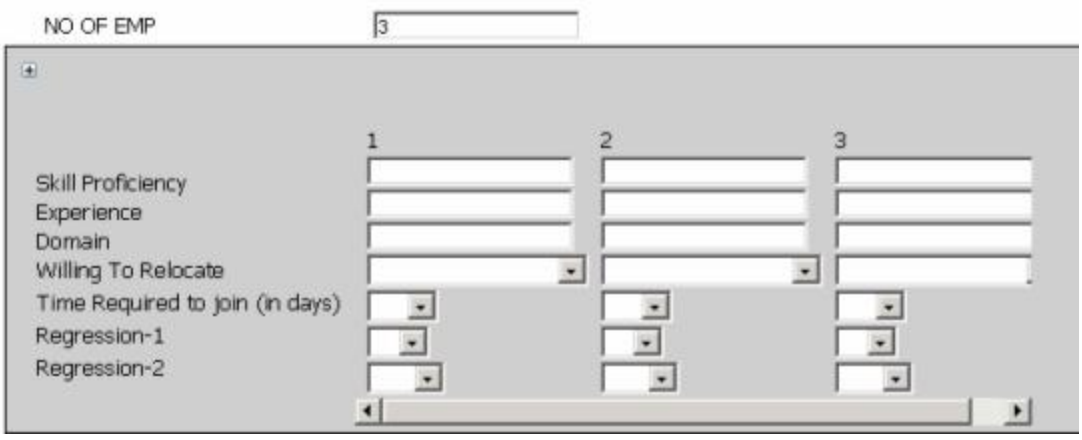
Setting Properties Tab

The Set Properties tab contains different options for editing the array attributes. In the Text Attribute Editor page:

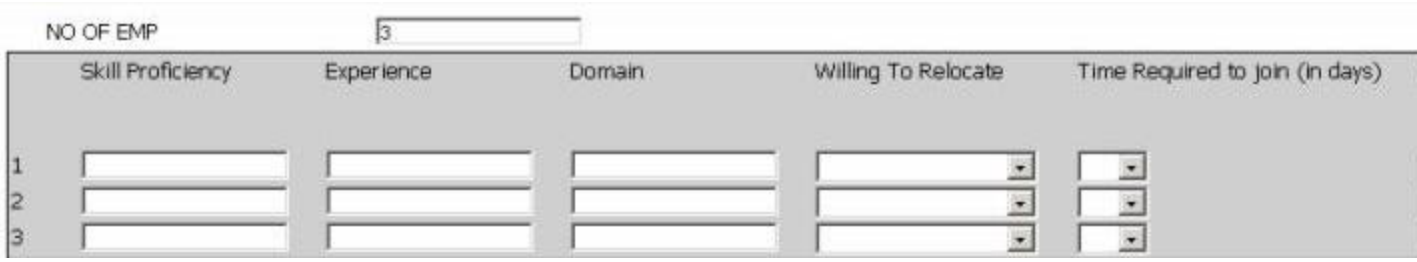
1. Navigate from Admin Home Page > Catalog Definition > Supported Products > Configurable Attributes > List > Click <Configurable Attribute of the type Array > Click Set Properties Tab
2. Select the attribute set from the navigation drop down menu against the Attribute Set field to modify the array set association.
3. Edit the name of the array attribute, if required. *The variable name of the array attribute is displayed as read-only.*
4. Enter the **description**. This is an optional field.
5. Choose the **Size Attribute f**or the array attribute.

> This drop down menu lists all the integer attributes available in the configured product, that have been marked as Array Control Attribute.

6. Choose the **display type of indexes** from the drop down menu against the Display Type field. The display types available in this list are:
   a. **Horizontal:** Horizontal display layout shows in the first row above each column



   b. **Vertical:** Vertical display layout shows indices in the first column of the table.



c. **Tabbed:** This layout shows the attributes for each array element in a separate tag.



7. Select the **display of labels** using the options available in the Label Option field:
    1. Display only First Label
    2. Display Both Labels and Indices (only allowed with Tabbed display layout)
    3. Display Only Labels (only allowed with Tabbed display layout)
8. Click **Apply** to remain on the current page, or click Update to return to the Attributes List Page.



The display properties of a set can only be controlled from the set properties tab and NOT from the template.

In the Set Association Information section, a list of the associated attributes is displayed. These are NOT editable. To add additional array attributes to the set, use the Array Attribute Editor page.

TROUBLESHOOTING

System throws an error

- If you try to declare an already existing control attribute as an attribute without the control property, then you are unable to do so if the if the attribute is referred as a size attribute to some set.
- If the default value of the control attribute is outside the allowed range.
- If a simple rule tries to set the value of the control attribute outside the specified range.
- If an advanced rule tries to set the value of the control attribute outside the specified range, the user on the buyside gets an error and buttons are disabled, forcing the user to not proceed. Changing the advanced rule enables these buttons once again.

NOTES



Create the array set BEFORE visiting the "Set" tab of the array attribute. Ensure the array control attribute is in the same group as the array set and verify it is not hidden.

Create arrays in this order: (1) Create the array control attribute, (2) Create the array set, (3) Create the array attributes, then (4) add the attributes to the array set. For more information on attribute ordering, see Configuration Attributes Overview.

See Array Best Practices for more information.



The rules run based on the attribute ordering on the Configurable Attributes Administration List page. Ensure that your Set Size Attributes are placed (ordered) before the Array Attributes for the rules to run effectively.

Configuration data in commerce is stored as a tilde delimited string. The array values within this string are delimited by $,$. For example, for a configuration array foobar that is of single-select menu type (possible values are red[r], green[g] or blue[b]), the string format looks like this &endash; foobar~-foobar~r$,$r$,$r$,$g. This means that foobar[0] = foobar[1] = foobar[2] = "r". foobar[3] = "g"



The Size Attribute must be marked as the Array Control Attribute upon attribute creation.

Multi-select menus are not possible in arrays.

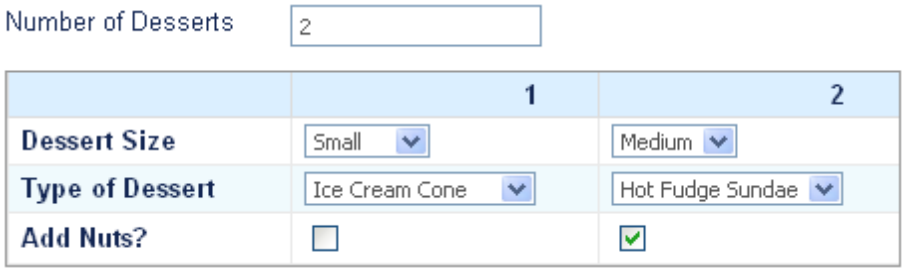
RELATED TOPICS

BOOLEAN ATTRIBUTES

OVERVIEW

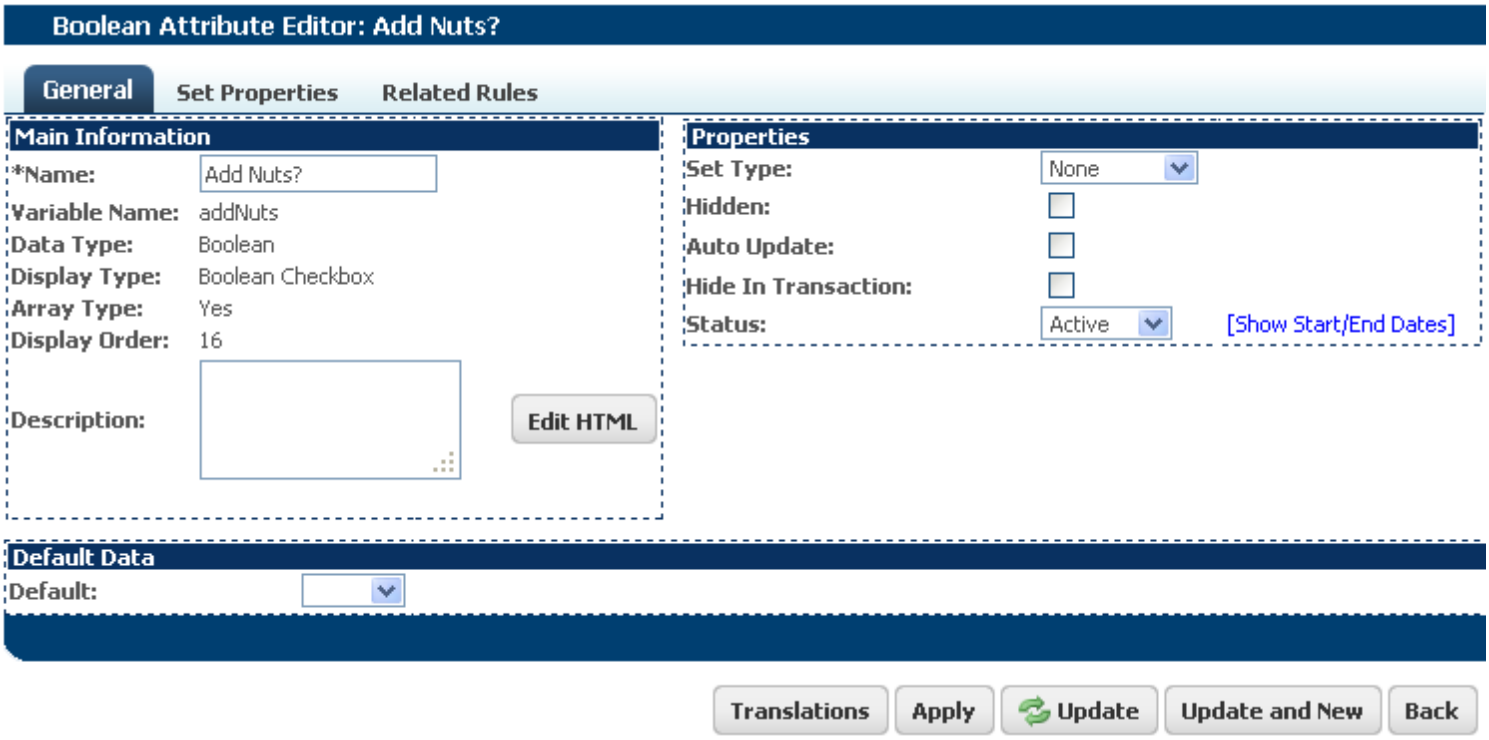
Boolean attributes allow you to define a field as any two choice option. Common uses are to indicate true/false, either/or, valid/invalid or to include/not include. Boolean attributes can be used for a variety of purposes. One use case would be to include a checkbox within an array, possibly for an add-on item.

Example of a Boolean Attribute in an Array



ADMINISTRATION

The Boolean Attribute Editor has three sections under the General Tab: Main Information, Properties and Default Data.



Managing Main Information

1. Edit the **Name** of the attribute.
2. Enter a **description**. *This will appear as the "help" information if the "Show Help Link" is checked in the configuration flow tab editor.*

Managing Properties

1. Choose the **Set Type**:
    1. **None:** The user can input any value without encountering an error message.
    2. **Set:** A recommendation rule can change the value entered by the user unless the attribute is locked by the user.
    3. **Forced Set:** A recommendation rule will change the value of the attributes regardless of what the user enters and if the attribute is locked or not.
2. Select the appropriate checkbox for:
    1. **Required:** This will require the user to enter a value before proceeding to a commerce transaction.
    2. **Hidden:** This attribute will not appear to the user. It can be used in rules.
    3. **Auto-Update:** This will by updated by the system without the user invoking an action.
    4. **Hide In Transaction:** This attribute is hidden within a commerce document.
3. Choose the **Status** of the attribute:
    1. **Active:** An attribute that is active will appear to the buyer and/or be included in rules. All attributes are active by default.
    2. **Inactive:** An inactive attribute cannot appear in a rule to the user.
    3. **Internal:** An internal attribute will only appear to fullaccessusers.
4. Click the **Show Start/End Date:**link to specify how long the attribute will remain in the status chosen.



If no duration is specified, the attribute is allowed to maintain the status chose until the admin changes. it.

Managing Default Data

1. Use the drop-down menu to select either True or False, if you wish the attribute to default to one of these selections.

USE CASES

Boolean attributes can be used in the condition, action, condition input and action input like other configurable attributes. There are a couple of things to note:

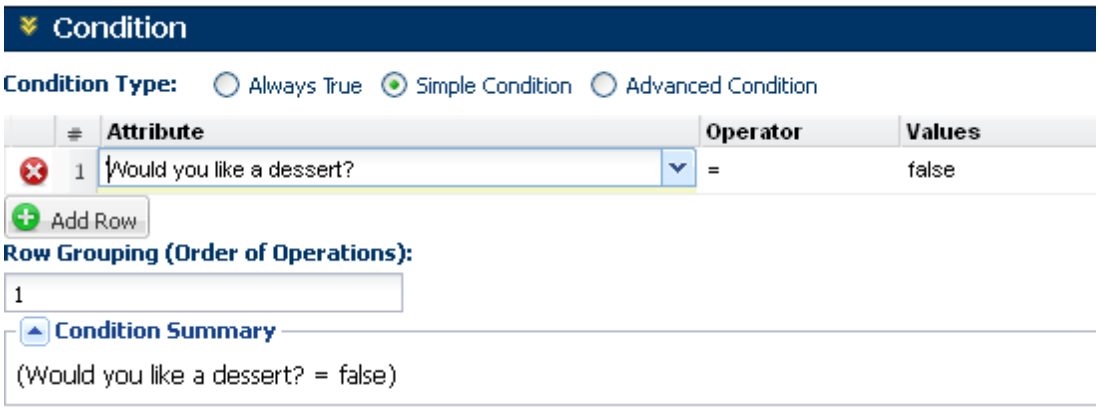
1. Boolean attributes can be used in both advanced and simple conditions.
2. When used in the simple condition, the following are Operators: "=" and "Not =".
3. Recommendations, Constraints, and Hiding rules an be defined on Boolean attributes.
4. Only simple action is available for constraint rules.
5. $BM_UNCHANGED_BOOLEAN$ is not supported for this release.

Adding a Boolean Attribute in a Rule

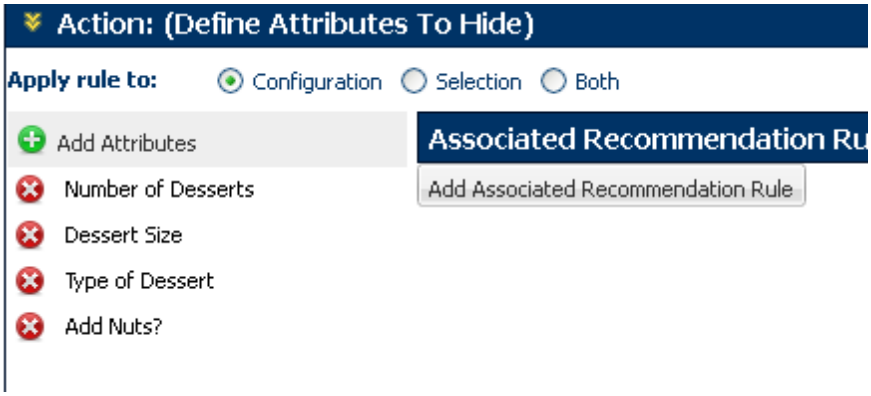
Here is a simple use case for adding a Boolean attribute in a rule. This assumes you have already created a Boolean attribute.

You don't want to display the desserts array on the UI unless Would You Like a Dessert = TRUE.

1. Navigate to Admin Home Page > Products > Catalog Definition > Product Families > Select a Product Family > Hiding Attribute
2. Click **Add**.
3. Enter a Name, Variable Name and Description.
4. Select the condition type: **Simple Condition**.
5. Click **Add Row** and enter the following:

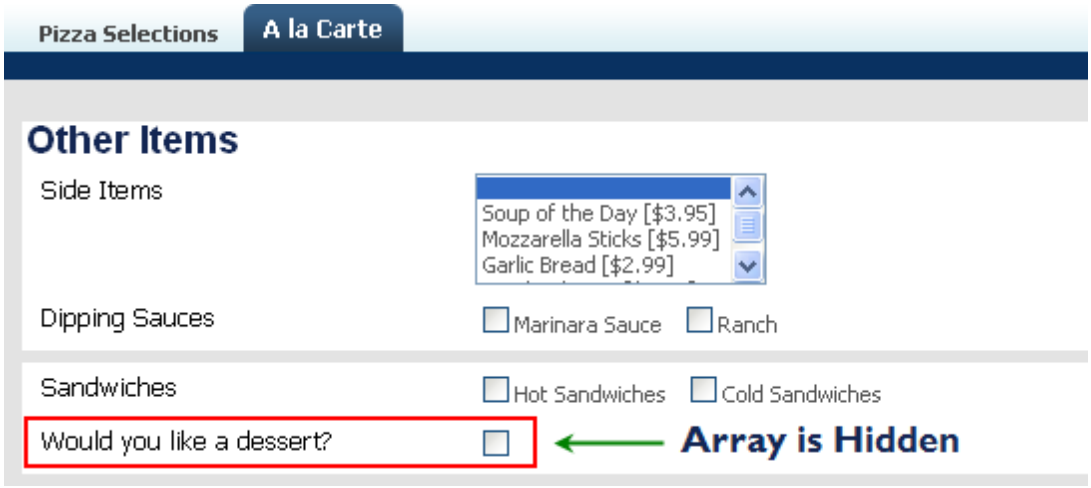


6. Make sure to set **Row Grouping**.
7. Add the **Actions** that you will hide when the condition is met.

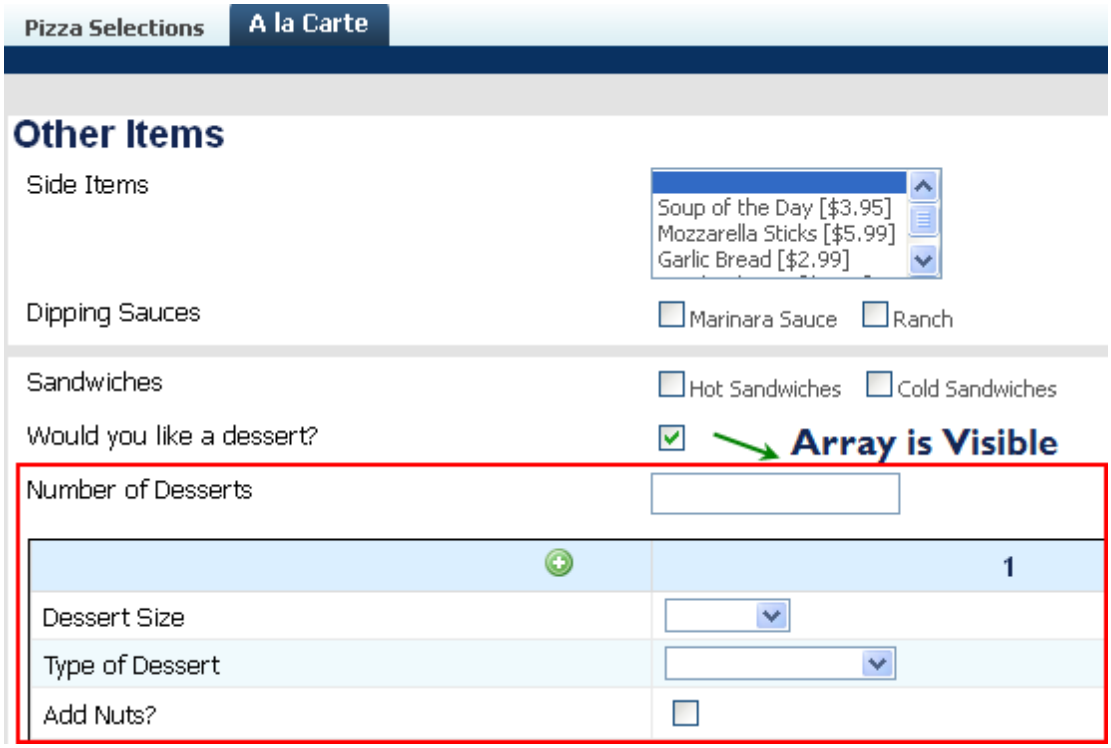


8. Click **Update**.

Example of the User-Side (Hidden Array - Condition is met)



Example of the User-Side (Visible Array - Condition is not met)



NOTES

When set as read-only, the Boolean attributes will display as a disabled checkbox.

Boolean attributes are displayed to the user as a checkbox; this functionality differs from Commerce when attributes are read-only.

There is not an option to hide/show a label.

RELATED TOPICS

CURRENCY ATTRIBUTES

OVERVIEW

The currency attribute is displayed as a floating point number field that will display the number in currency format. The currency displayed is the user's currency. Currency attributes can be of array and non-array types. A range check can be defined on currency attributes. Currency is also supported for single-select and multi-select menus.



Configuration currency attributes support negative currency. However, no special fomatting is available for negative currency. Currency is formatted based on the current user's currency preference.

ADMINISTRATION

The Text Attribute Editor has three sections under the General Tab: Main Information, Properties and Default Data.

Main Information

1. Edit the **Name** of the attribute.
2. Enter a **description**. *This will appear as the "help" information if the "Show Help Link" is checked in the configuration flow tab editor.*
3. Enter **default data** that will populate the field (if desired).



There is an option for Range Check validation.

Properties

1. Choose the **Set Type**:
    1. **None:** The user can input any value without encountering an error message.
    2. **Set:** A recommendation rule can change the value entered by the user unless the attribute is locked by the user.
    3. **Forced Set:** A recommendation rule will change the value of the attributes regardless of what the user enters and if the attribute is locked or not.
2. Select the appropriate checkbox for:
    1. **Required:** This will require the user to enter a value before proceeding to a commerce transaction.
    2. **Hidden:** This attribute will not appear to the user. It can be used in rules.

3. **Auto-Update:** This will by updated by the system without the user invoking an action.
4. **Hide In Transaction:** This attribute is hidden within a commerce document.

3. Choose the **Status** of the attribute:
    1. **Active:** An attribute that is active will appear to the buyer and/or be included in rules. All attributes are active by default.
    2. **Inactive:** An inactive attribute cannot appear in a rule to the user.
    3. **Internal:** An internal attribute will only appear to fullaccessusers.
4. Click the **Show Start/End Date:**link to specify how long the attribute will remain in the status chosen



If no duration is specified, the attribute is allowed to maintain the status chosen until the admin changes it.

5. Select decimal precision, based on currency, in the Currency Properties.

USE CASES

Use in Configuration Rules

Currency attributes can be used in the Condition, Action, Condition Input and Action Input like other configurable attributes. In advanced functions, currency is treated as a float.

- Currency attributes in Condition – Currency attributes can be used in the Simple and the advanced condition.
    - When used in the Simple Condition, the following options are available as Operators: “=”, “Not =”, “>”, “<”, “>=”, “<=”
    - The simple condition grid also supports ranges for currency attributes.
- Currency Attributes in Action – Recommendations, Constraints and Hiding rules can be defined on currency attributes.
    - Simple action for constraint rules allow the fullaccessuser to constrain a range of values for currency attributes
    - $BM_UNCHANGED_NUM$ is supported for array type currency attributes



For a multi-currency application, if the user currency is not the same as the base currency, then exchange rates are applied to convert the currency attribute value to the corresponding value in the base currency for use in BML. Anything returned from BML is in base currency and exchange rates applied on the way back out.

Use in Configuration Pages

Configuration Flow

Currency Attributes can be displayed in the Default and Advanced Templates.

Search Flow

Currency can be displayed in the Search Flow in the Default or Advanced Template

- Currency in Function to evaluate Result Set – Only unformatted currency is passed into the Function to evaluate Result Set. The currency will be formatted on the Search Result page based on the current user's session currency.
- Currency in popup function – Formatted currency must be passed into popup function as currency formats will not be applied on the end user side.

Currency attributes in BML and BML Library

The following functions have been provided for currency attributes:

- **formatascurrency:** This function takes a number and returns it as a formatted string
  - **Syntax:** String formatascurrency(Float x, [String currencyCode]
  - **Example**:formatascurrency(32.15, "EUR"); will return the string "€32,15".
- **getcurrencyvalue:** This function takes a formatted currency string and returns the numeric value
  - **Syntax**: Float getcurrencyvalue(String value, [String currencyCode])
  - **Example:** getcurrencyvalue("€32,15", "EUR"); will return the float 32.15

NOTES



Currency attributes are displayed in the format of the session currency of the user in configuration.



The fullaccessuser can choose to display currency in the printed document using the formatting functions available.

Currency values are carried from configuration to commerce and can be displayed on the sub-document in the format of the document currency.

Zero and blank integer values are treated as separate values:



-Zero = 0
-Blank = ""



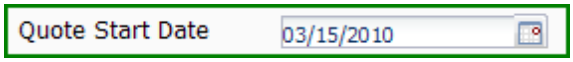
Currency values in SOAP are not formatted through the user's session currency is taken into account to apply exchange rates and return the correct values.

RELATED TOPICS

DATE ATTRIBUTES

OVERVIEW

The Date type attribute lets you define and use dates in search and configuration flows. Date attributes can be created at all levels of the product hierarchy, just like other configurable attributes. All attribute properties can be defined as they would on a text field type configurable attribute, including Set Type, Auto-Update, Hidden, Required, Hide in Transaction and Status. Another major feature is the ability to define a Range Check on dates.



The date attribute is displayed as a text box with a date selector tool. This gives the user the option to either enter a date manually or select one from the calendar. There is an automatic validation that runs to ensure that the date complies with the application and user profile information. Date attributes can be of array or non-array type. Date arrays cannot be displayed as menus.

ADMINISTRATION

The Date Attribute Editor has three sections under the General Tab: Main Information, Properties and Default Data.

Main Information

1. Edit the **Name** of the attribute.
2. Enter a **description**. *This will appear as the "help" information if the "Show Help Link" is checked in the configuration flow tab editor.*

Properties

1. Choose the **Set Type**:
    1. **None:** The user can input any value without encountering an error message.
    2. **Set:** A recommendation rule can change the value entered by the user unless the attribute is locked by the user.
    3. **Forced Set:** A recommendation rule will change the value of the attributes regardless of what the user enters and if the attribute is locked or not.
2. Select the appropriate checkbox for:
    1. **Required:** This will require the user to enter a value before proceeding to a commerce transaction.
    2. **Hidden:** This attribute will not appear to the user. It can be used in rules.
    3. **Auto-Update:** This will by updated by the system without the user invoking an action.
    4. **Hide In Transaction:** This attribute is hidden within a commerce document.

3. Choose the **Status** of the attribute:
    1. **Active:** An attribute that is active will appear to the buyer and/or be included in rules. All attributes are active by default.
    2. **Inactive:** An inactive attribute cannot appear in a rule to the user.
    3. **Internal:** An internal attribute will only appear to fullaccessusers.
4. Click the **Show Start/End Date:**link to specify how long the attribute will remain in the status chosen



If no duration is specified, the attribute is allowed to maintain the status chosen until the admin changes it.

Default Data

1. Enter a default data or select a date from the date picker (if necessary).
2. Select the box "Use Current Date" if you wish to always use the current date as the default data.
3. Select a validation, if necessary:
    1. **None:** A validation will not be run.
    2. **Range Check**: Will allow you to enter a date manually, select one from the calendar tool or use the current date. This can be used to verify that a date falls within an appropriate range before proceeding.

USE CASES

Use in Configuration Rules

Users will be able to use Date Attributes as simple and advanced condition attributes within all configuration rules. Dates can be used in the Condition, Action, Condition Input and Action Input like other configurable attributes. In addition, Recommendations and Constraints defined on Date Attributes return Date objects.

The following operators are available when being used in simple conditions:

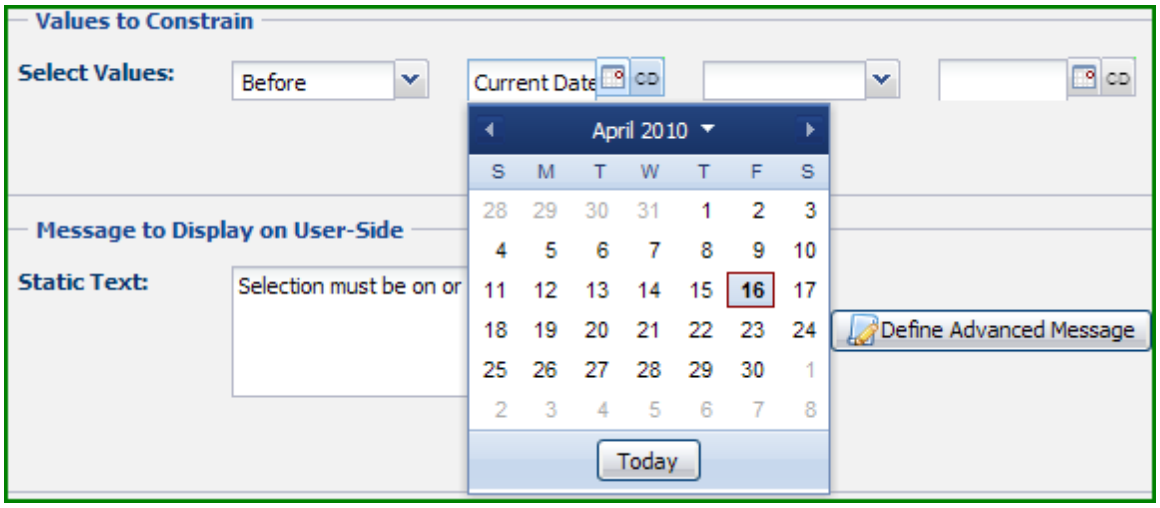
- =(equal)
- Not=
- Before
- On or Before
- After
- On or After

Recommendation Rules

Recommendations rules are used to recommend values for non-array and array dates. Set type for recommendations work the same way for dates as they do for all other configurable attributes. User can use date attributes within Recommendation rules which can be used to set the value of date attributes. Uses can do this by either entering date values or using BML. If using BML, the return data type would be: date.

Constraint Rules

If a date is constrained, it will be down as grayed-out in the date picker and the system should throw a constraint message. To create a simple constraint on a date attribute, the user can enter a single date or a date range to constrain:



In order to constrain multiple dates, the user will have to define a BML function and return an array with a new data type: Date[]. When doing this, every date within the array will be constrained on the user-side.

Configuration and Search Flows

In configuration and/or search flows, dates can be displayed in the default and advanced templates.

1. Dates in Function to return a Result Set – In the function to evaluate result set, the Date attribute has to be passed in the db date format. (string format)

```
//Snippet for passing dates into the Function to return a Result Set
str = str + "<AttributeVariableName=\"simpleDate"\"> + (datetoStr(simpleDate, "yyyy-MM-dd hh:mm:ss"));
```

2. Array Dates in Functions to return a Result Set - In the fuction to evaluate result set, the date array has to be passed into the getarrayattrstring function.

```
//Snippet for passing date arrays in the Function to return a Result Set
str = str + "<Attribute VariableName=\" arrayDateCurrentDateAsDefault\>'
+ getarrayattrstring(array DateCurrentDateAsDefault) + "<\Attribute>";
```

NOTES



Blank dates displayed to the end user are treated as null values. It is recommended that while reading values of date attributes, the fullaccessuser performs an isnull check to avoid errors.



Date attributes are supported in Bulk Upload and Download.

Date attributes are supported in Configuration SOAP API in the database format [yyyy-MM-dd 00:00:00]. The user cannot set a time component for dates through SOAP. All dates are set with the time compoenent as 00:00:00.

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""



Dates in configuration DO NOT support time.

RELATED TOPICS

HTML ATTRIBUTES

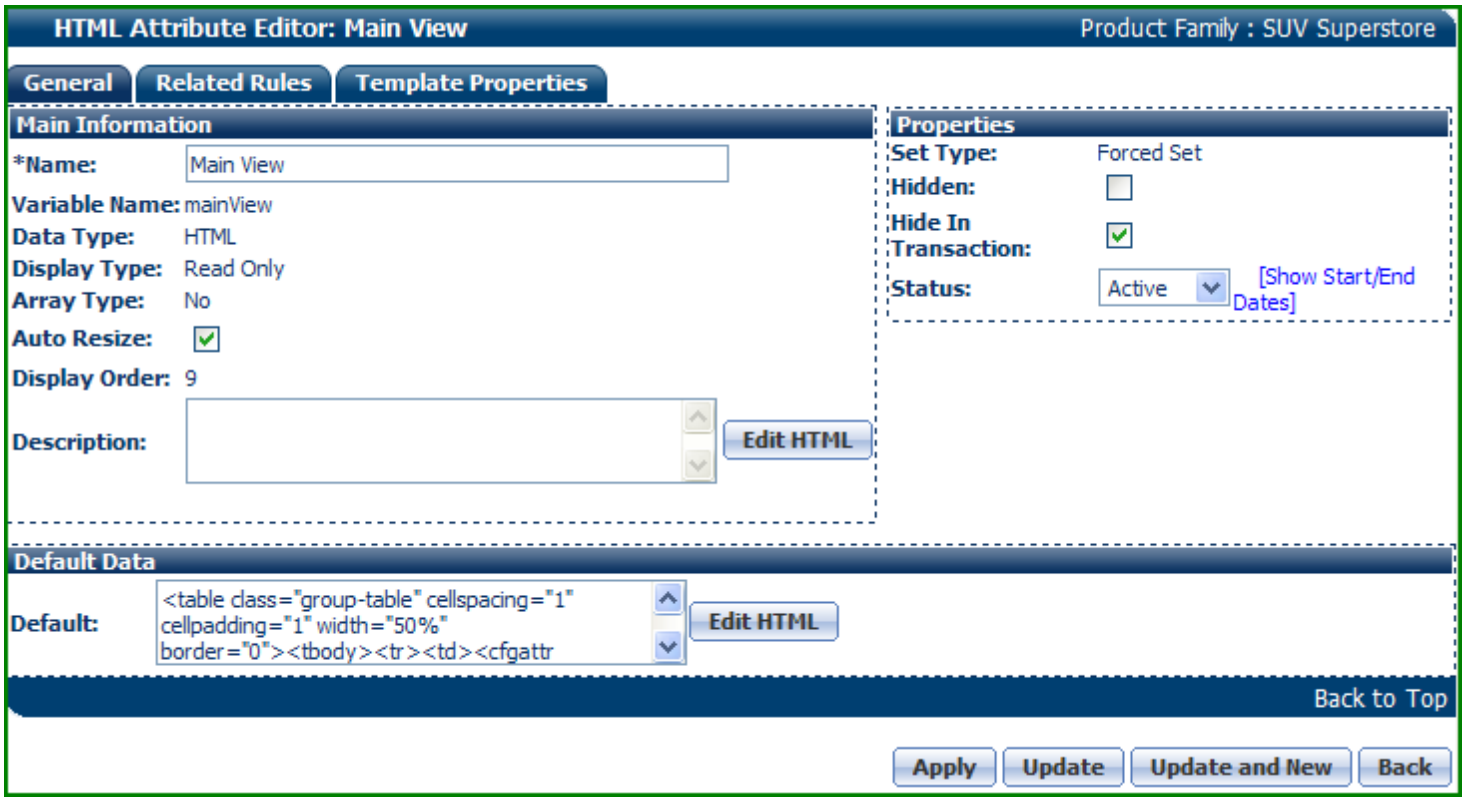
OVERVIEW

You can insert customer HTML in a configuration page by defining an HTML type attribute at any level in the product hierarchy. These attributes can be associated with default or custom templates in the configuration or selection flows. HTML attributes can be used as action attributes in Recommendation and Hiding rules. In Recommendation rules, the must be of forced set type.

Configurable HTML attributes are carried to commerce like other configurable attributes. They can be displayed in the transaction as well as in the printer-friendly documents.

ADMINISTRATION

The HTML Attribute Editor has three sections under the General Tab: Main Information, Default Data and Properties.



Managing Main Information

1. Edit the **Name** of the attribute. *The variable name cannot be changed.*
2. Select the **Auto Resize** option to enable the attribute to fit all of its content on the user side.



If it is unchecked, then you can give the attribute dimensions (in pixels) and it will remain that size. Scorllbars will appear if there is overflow content.

3. Enter a **description**. *This will appear as the "help" information if the "Show Help Link" is checked in the configuration flow tab editor.*

Managing Default Data

1. Click the Edit HTML box in order to edit the HTML source for the attribute. See Data Types of Configurable Attributes for a sample use case.

Managing Properties

1. Notice that the Set Type cannot be edited. It is always a Forced Set.
2. Select the appropriate checkbox for:
    1. **Hidden:** The attribute will not appear to the user. It can be used in rules. By default, HTML attributes are marked as hidden.
    2. **Hide In Transaction**: The attribute is hidden within a commerce document.
3. Choose the**Status** of the attribute:
    1. **Active:** An attribute that is active will appear to the buyer and/or be included in rules. All attributes are active by default.
    2. **Inactive:** An inactive attribute cannot appear in a rule to the user.
    3. **Internal:** An internal attribute will only appear to fullaccessusers.
4. Click the **Show Start/End Date:**link to specify how long the attribute will remain in the status chosen

---

Printer-Friendly

The configuration HTML attributes can be displayed in the printer-friendly document. They need to be looped over the line items for it to be printed per line.

If you are using table structures in configurable HTML attributes and want to render them correctly in the PDF, then your table tags must contain the column attributes which contains the number of table columns and the corresponding widths as follows:

The above is the default structure of the table generated when a table is inserted in the HTML editor. However, if the fullaccessuser uses the source code mode in the HTML editor and codes the table tag, they must remember to manually put in the columns attribute. Otherwise, the table would fail to render correctly on the PDF.

The unit of the column width is set in inches in the parser. The table structure received from the configuration HTML editor has inches hard-coded into the table creation window so it does not allow other units for column width.

USE CASES

Changing an image based on menu selection

Let's say you would like to dynamically change an image based on a menu selection.

1. Create an **HTML attribute**.
2. Select the **Hidden** checkbox under Properties.
3. Create a recommendation rule. Use your new HTML attribute as the action attribute.



In Recommendation rules, HTLM attributes must be of forced-set type.

4. Create your advanced function designed to return the proper image based on your menu selections.

```
baseStr = "http://yourapp.bigmachines.com/bmfsweb/yourapp/image/folder";
imageParams = "";

if (chooseAnExteriorColor == "Satin White Pearl"){
        imageParams = imageParams + "satin_white_pearl";
        }

return baseStr + "/" + imageParams + ".png";
```

5. Return to your configurable attribute and use both HTML and JavaScript to dynamically change the image.
    1. **HTML Script:**

```
<table class="group-table" cellspacing="1" cellpadding="1" width="50%" border="0">
<tbody>
    <tr>
        <td><cfgattr _variablename="mainViewLink" _hidden="YES" _display="INPUT_FIELD" />
        </cfgattr />
        <center><img id="imageLink" height="200" src="" name="imageLink" alt="" />
        </center>
        </td>
    </tr>
</tbody>
</table>
```

2. **Sample JavaScript code:**

```
<script>
var link = "";
    if (getHAttrs("mainViewLink")) {
        link = getHAttrs("mainViewLink").initialValues[0];
        document.getElementById("imageLink").src = link;
}
</script>
```

Now that this has all been created, you should be able to see your image/images when you are on the configuration page:



NOTES

Date and HTML attributes cannot be menus. Also, HTML attributes cannot be used in an array.

HTML attributes are used as strings when debugging Recommendation rules.

HTML attributes are marked as 'hide in transaction' by default, for performance reasons. Deselect this if you need to use them in Commerce or the Document Engine.

Content of the HTML attributes is rendered as un-prased text in the Commerce sub-document.



Ill-constructed HTML can break Configuration Layouts, so the fullaccessuser must be very careful about setting values for HTML fields.

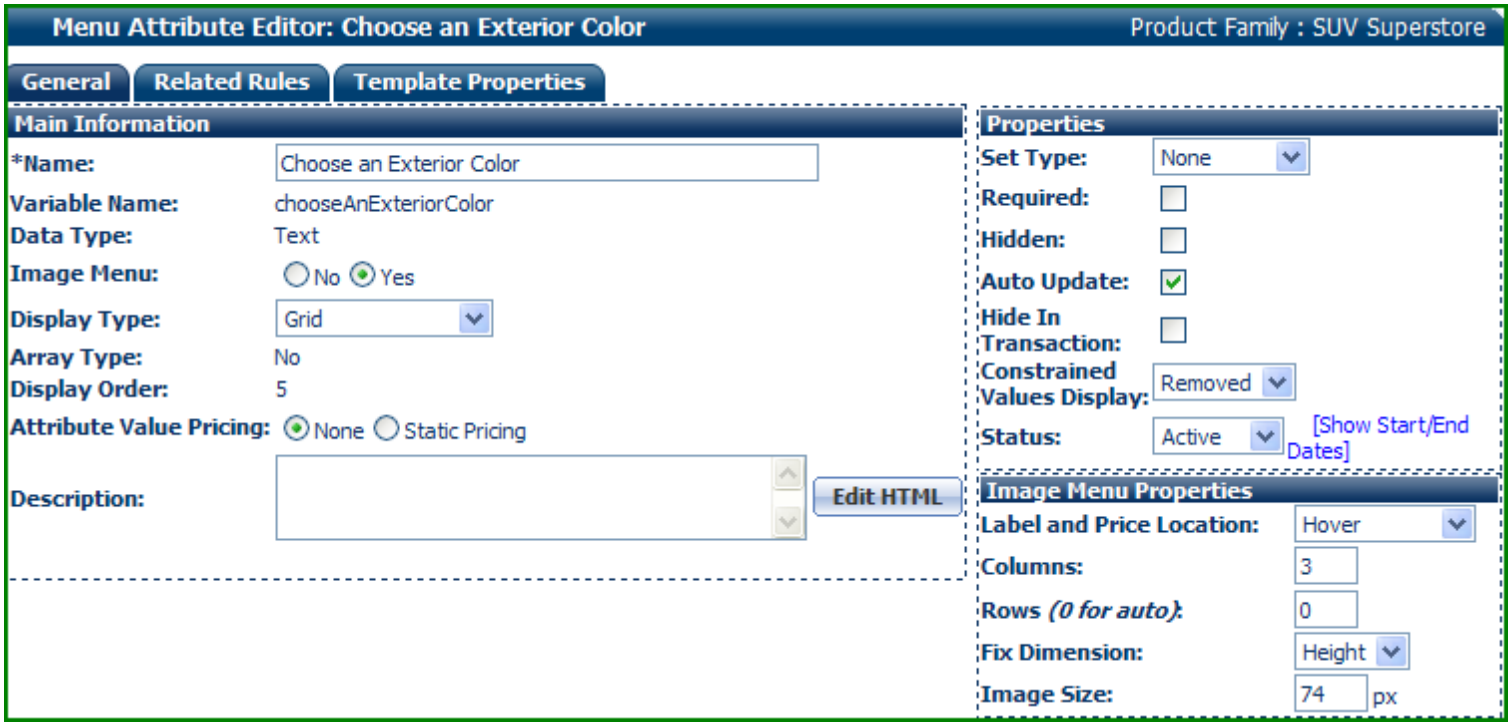
HTML attributes cannot be used as a rule condition or in configuration arrays.

RELATED TOPICS

MENU ATTRIBUTES

OVERVIEW

The Menu Attribute Editor has three sections under the General Tab: Main Information, Properties and Using Images with Attributes Properties.



ADMINISTRATION

Managing Main Information

1. Edit the **Name** of the attribute.
2. Select **Yes** to enable Image Menus, if desired. See Image Menus for more details.
3. Choose the **Display Type**:
    1. **Single Select Menu Options**: Single Select Drop-Down Menu, Radio Button Vertical, or Radio Button Horizontal.
    2. **multi-select Menu Options**: multi-select Drop-Down Menu, Checkbox Vertical, or Checkbox Horizontal.
4. Select **Static Pricing** to enable Attribute Value Pricing functionality. *The default selection is None.*
5. Enter a **description**. *This will appear as the "help" information if the "Show Help Link" is checked in the configuration flow tab editor.*

Managing Properties

1. Select the **Set Type**:
    1. **None:** The user can input any value without encountering an error message.
    2. **Set:** A recommendation rule can change the value entered by the user unless the attribute is locked by the user.
    3. **Forced Set:** A recommendation rule will change the value of the attributes regardless of what the user enters and if the attribute is locked or not.
2. Select the appropriate checkbox for:
    1. **Required:** This will require the user to enter a value before proceeding to a commerce transaction.
    2. **Hidden:** This attribute will not appear to the user. It can be used in rules.
    3. **Auto-Update:** This will by updated by the system without the user invoking an action.
    4. **Hide In Transaction:** This attribute is hidden within a commerce document.
3. Choose the **Constrained Value Display**:
    1. **Removed:** Constrained attribute values will be removed from the menu.
    2. **Marked:** Constrained attribute values will be marked with an asterisk (*) in the menu.
    3. **Normal:** Constrained attribute values will appear in menus without being marked.
4. Choose the **Status** of the attribute:
    1. **Active:** An attribute that is active will appear to the buyer and/or be included in rules. All attributes are active by default.
    2. **Inactive:** An inactive attribute cannot appear in a rule to the user.
    3. **Internal:** An internal attribute will only appear to fullaccessusers.
5. Click the **Show Start/End Date:**link to specify how long the attribute will remain in the status chosen



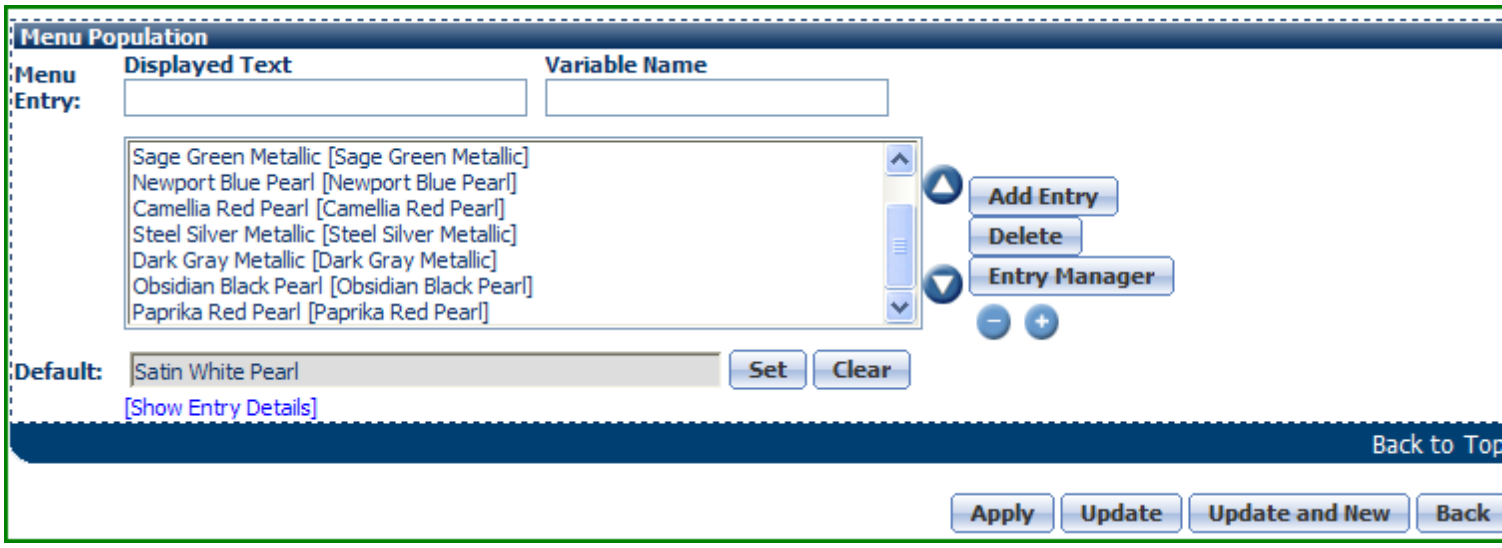
If no duration is specified, the attribute is allowed to maintain the status chosen until the admin changes it.

Managing Menu Population

At the bottom of the General Tab, there will be an area available for you to populate your menu.

1. Enter a **Displayed Text Value.**
2. Enter a **Variable Name**.

3. Click **Add Entry**



4. Click **Entry Manager** or **Show Entry Details** to edit the order of the menu attribute, change the menu display names, add prices or add images to each menu value.

5. Set a **Default** menu value by selecting a menu value and **clicking Set**.

RELATED TOPICS

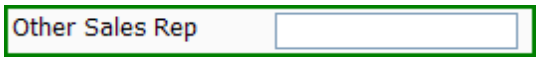
TEXT, FLOAT, AND INTEGER ATTRIBUTES

OVERVIEW

Configurable attributes can be created as menus or text fields. Text fields are more open-ended allowing the buyer to type in values. You also can choose the status of the configurable attributes as active, inactive or internal. Configurable attributes can also be made active during certain time periods by specifying start and end date.

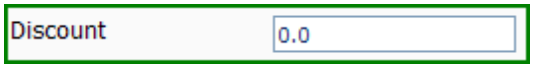
Text Attributes

Displays a text field or text area that the user can populate with data. This can be displayed as a text field or as a single-select or multi-select menu.



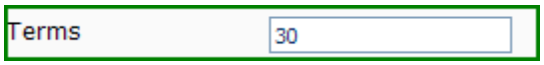
Float Attributes

Floating point number fields validate numbers with decimal values. This can be displayed as a text field or as a single-select or multi-select menu.



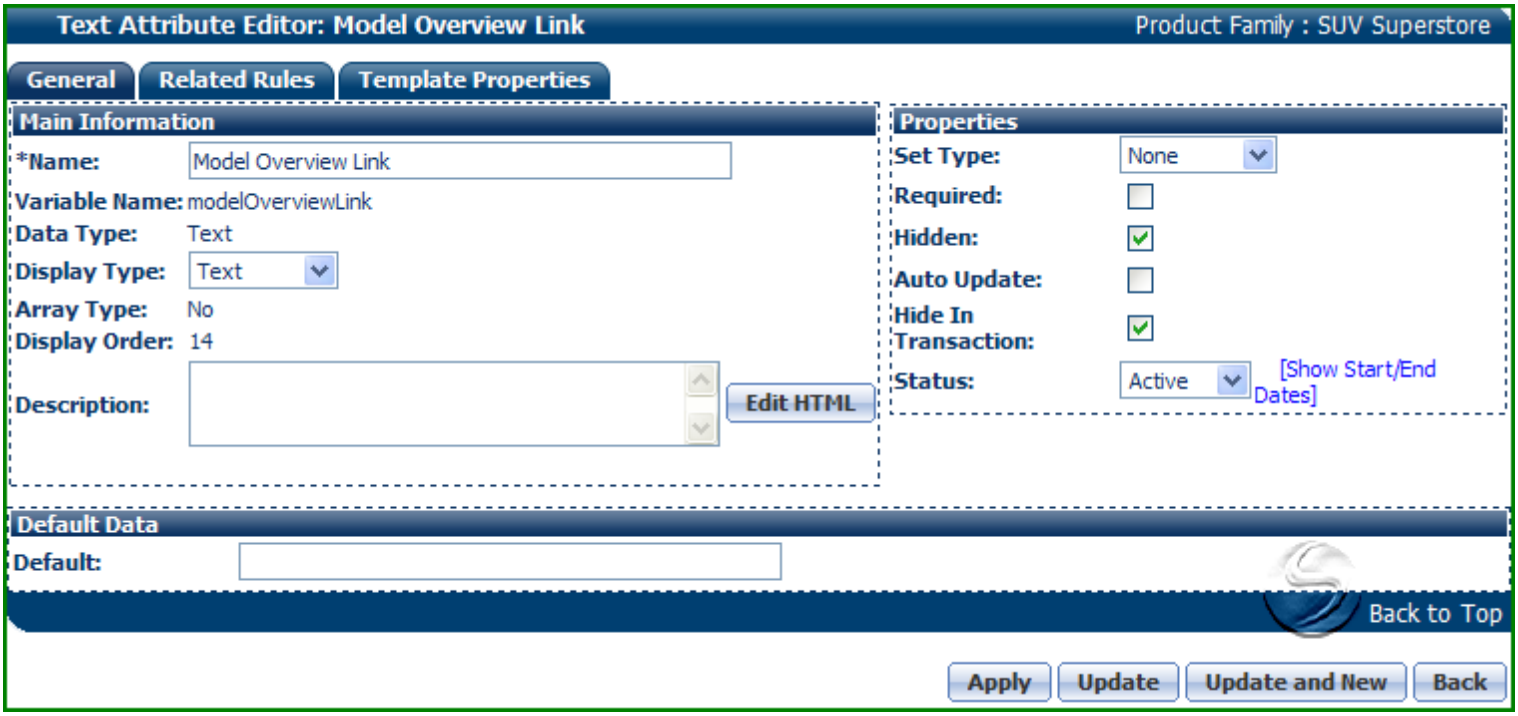
Integer Attributes

Integer number fields validate that only whole numbers are entered. This can be displayed as a text field or as a single-select or multi-select menu.



ADMINISTRATION

The Text Attribute Editor has three sections under the General Tab: Main Information, Properties and Default Data.



Managing Main Information

1. Edit the **Name** of the attribute.
2. Choose the **Display Type**: Text or Text Area. *The Text Area allows you to allow more than 255 characters in the field*
3. Select **Array Control Attribute**, if desired. This option only appears for Integer type attributes. See Managing Arrays and Sets for more information.
4. Enter a **description**. *This will appear as the "help" information if the "Show Help Link" is checked in the configuration flow tab editor.*
5. Enter **default data** that will populate the field (if desired).

> Float and Integer type attributes will also have an option for a Range Check or Positive Number validation.

Managing Properties

1. Choose the **Set Type**:
    1. **None:** The user can input any value without encountering an error message.
    2. **Set:** A recommendation rule can change the value entered by the user unless the attribute is locked by the user.
    3. **Forced Set:** A recommendation rule will change the value of the attributes regardless of what the user enters and if the attribute is locked or not.
2. Select the appropriate checkbox for:
    1. **Required:** This will require the user to enter a value before proceeding to a commerce transaction.
    2. **Hidden:** This attribute will not appear to the user. It can be used in rules.
    3. **Auto-Update:** This will by updated by the system without the user invoking an action.
    4. **Hide In Transaction:** This attribute is hidden within a commerce document.
3. Choose the **Status** of the attribute:
    1. **Active:** An attribute that is active will appear to the buyer and/or be included in rules. All attributes are active by default.
    2. **Inactive:** An inactive attribute cannot appear in a rule to the user.
    3. **Internal:** An internal attribute will only appear to fullaccessusers.
4. Click the **Show Start/End Date:**link to specify how long the attribute will remain in the status chosen



If no duration is specified, the attribute is allowed to maintain the status chosen until the admin changes it.

NOTES



Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""

RELATED TOPICS

USING IMAGES WITH ATTRIBUTES

OVERVIEW

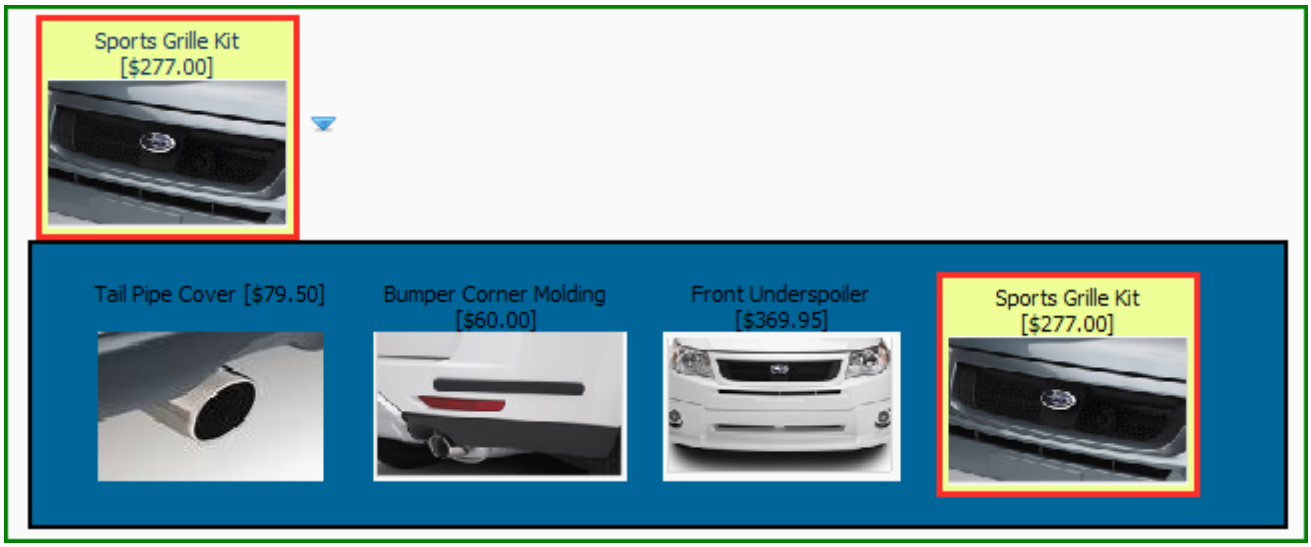
BigMachines allows you to display images next to attribute as well as show menu values as images instead of text. This allows users to choose a menu value by clicking on the image. Image Menus are available for both array and non-array type single-select and multi-select menus.

If you chose to populate a menu with images, you can choose to display them as either a Grid or a Drop-Down menu:

Example of a Grid Menu



Example of a Drop-Down Menu



ADMINISTRATION

Before you being with Image Menus, it is recommended that you update the Configuration StyleSheet. Simply navigate to All Product Families > Style sheets and click Update.

Adding Image Menus

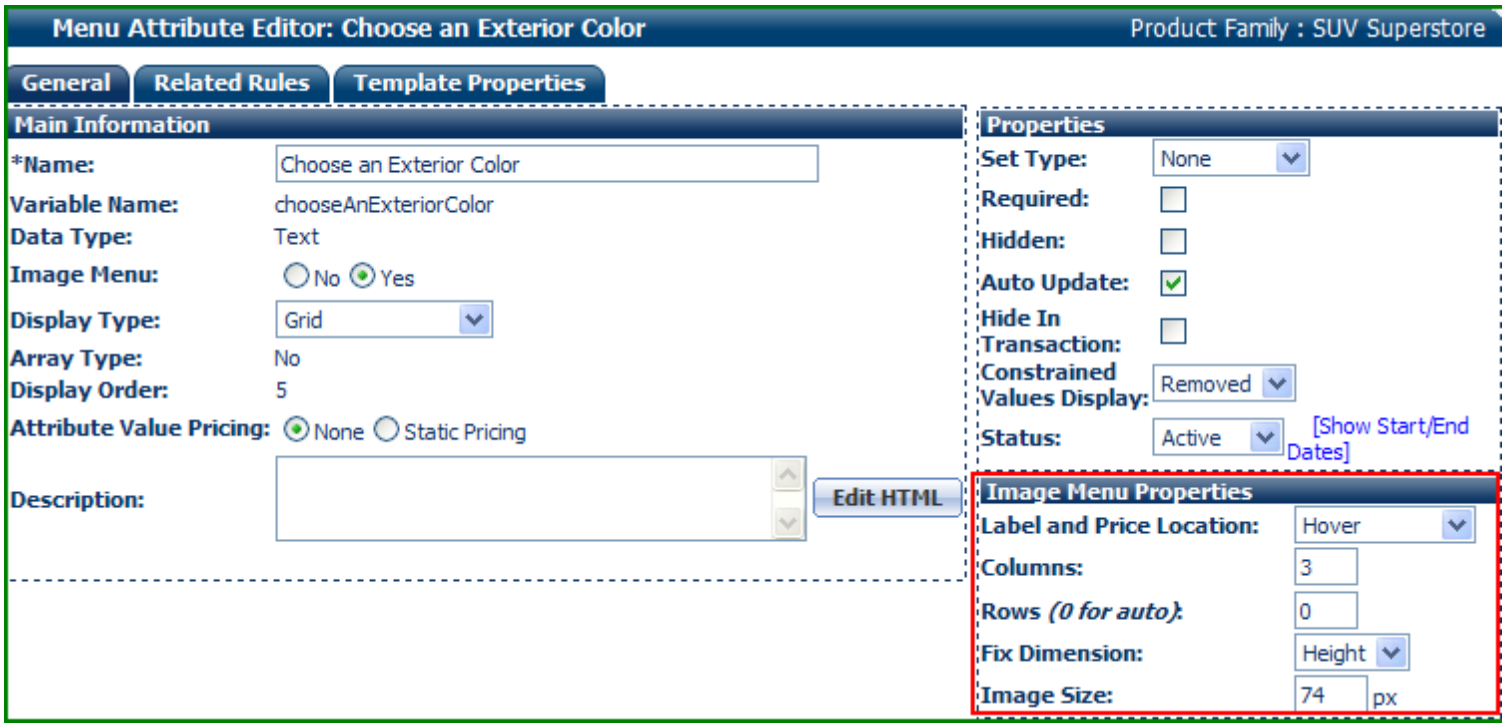
1. Navigate to a **Configurable Attributes List** page and select the attribute to which you want to add images.
2. **Select Yes** for the Image Menu option.

> If you leave this option as No, you can still add images to the menu, but they will display next to your configurable attribute.

3. Choose the **Display Type**: Grid or Drop-down Menu. *See Overview for examples*.
4. Click the **Entry Manager**.
5. Click **Browse** to search the File Manager and select the appropriate image you'd like to correspond with your menu value or click Clear to clear an existing image.

> Make sure your images have been uploaded to the file manager.

6. Save and close the Entry Manager. You should see a section labeled **Image Menu Properties.**



7. Set the following:

   a. **Label and Price Location**: Determine if where you'd like to display the label and attribute value price(if being used) of each menu item: When Hovering Over Image, Shown Above the Image or Shown Below the Image.



   b. **Rows and Columns:** Determine how many rows and columns you'd like to display the images in.

> If you select 0 rows, the number of rows will be auto-populated based on the number of columns you have selected. For example, if you have 12 menu items and you'd like to display them in rows of 3, you would need to enter 3 for Columns. Now, if you wanted to make sure that instead of displaying all 4 rows, that only the first 2 rows appear, make sure to enter 2 rows. The image menu will then display a scroll bar so you can see the rest of the rows.



   c. **Fix Dimension**: Base the image size on height or width. The images will be auto-scaled to preserve the proportions of the image to fit the layout.

   d. **Image Size**: Determine the image size in pixels.



It is best practice to create your images in a size similar to how you will have them be displayed in order to keep the image dimensions consistent and avoid distortion.

NOTES



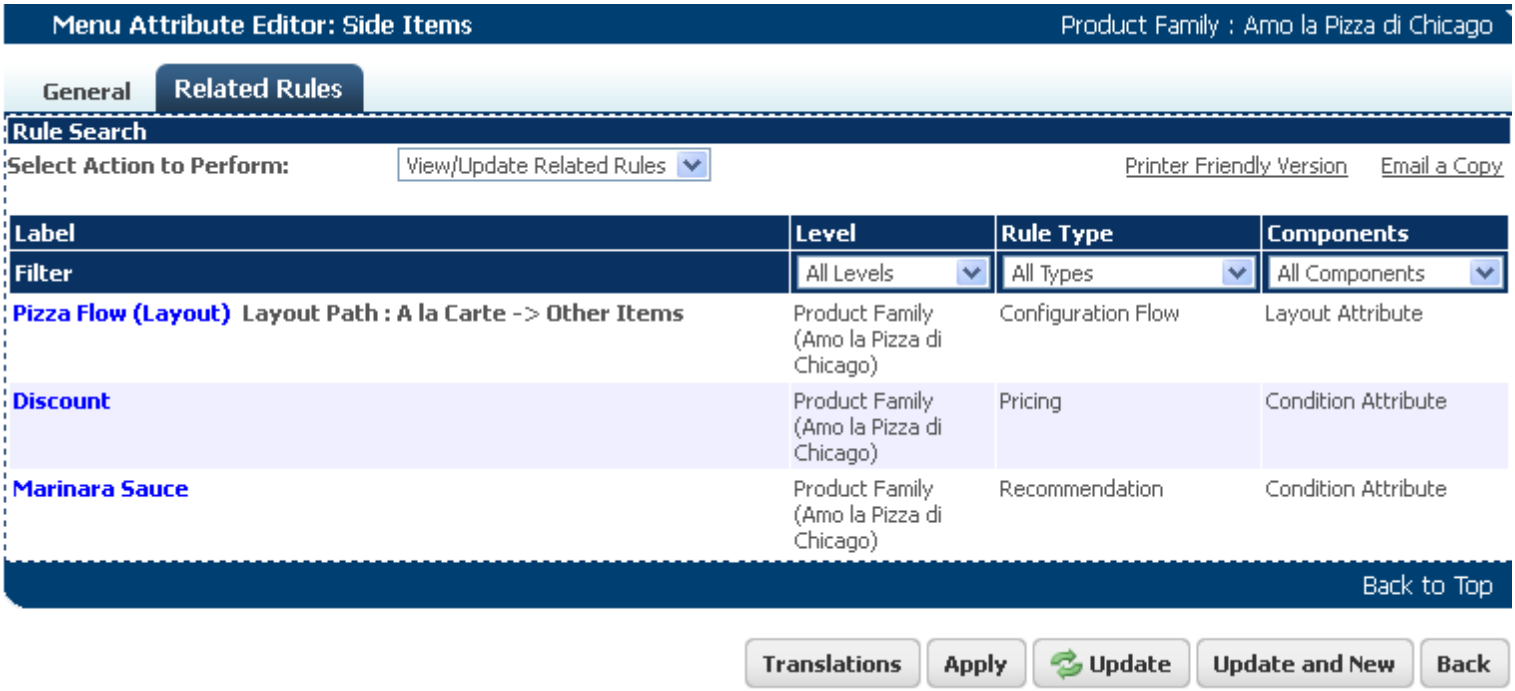
You can style Image Menus by providing style options in either the Style Sheet Editor for a default style sheet or by uploading an alternate style sheet.

RELATED TOPICS

RELATED RULES TAB

OVERVIEW

In the related rules tab, you can view all of the rules that are using this particular attribute. It will tell you the name of the rule, what level its located at in the hierarchy, the rule type and the components.



Rule Search

In this section, there is a drop-down menu for Select Action to Perform. You can select from the drop-down menu:

- **View/Update Related Rules**: This will populate a list of rules that this attribute is related to. you can click on each rule if you wish to make changes.
- **View/Update All Rules**: This expands the list to include all rules that occur at all levels within a Product Family. You can use the Level menu to drill down through Product Family, Product Line and Model level rules.
- **Create New Rule:** Selecting this action will allow you to create a new rule and open the appropriate rule editor. In order to do so:
    1. Select the level where this rule will fire. Product Family is already selected, but you can choose Product Line and Model if you wish.
    2. Select the Rule type: Recommendation, Constraint, Bill of Material, Pricing, Configuration Flow, Recommended Items or Hiding Attributes.
    3. Click Create to open the Rule Editor

Filter

Under the Filter section, you can see the list of rules with which the attribute is associated. You can click on the rule to pop-up a window with that rule's editor.

RELATED TOPICS

ATTRIBUTE VALUE PRICING (AVP)

OVERVIEW

Attribute Value Pricing provides the fullaccessuser with an easy way of maintaining pricing instead of creating a pricing rule for each attribute. This feature allows the fullaccessuser to associate static prices with values of a **menu type** configurable attribute so that these prices can be displayed to the user upon selection of the menu values on the model configuration page. A unique aspect of attribute value pricing is the ability to update price totals of a configurable item instantly without refreshing the page.

AVP highlights

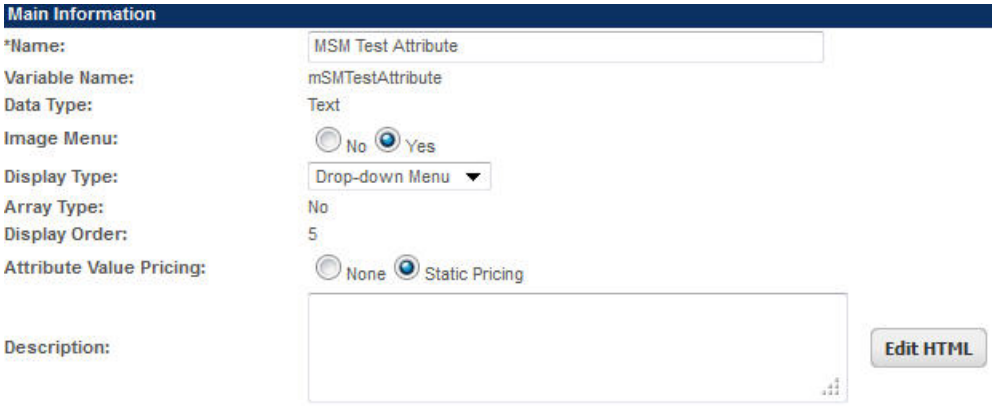
1. AVP is available for multi-select and single-select attributes for all data types including array type single-select menu attributes.
2. Static prices affect the final price of a product much like pricing rules and are also carried over as component prices to commerce.
3. On the user side, the static prices can be displayed in the pricing section along with other pricing rules. The fullaccessuser can choose to include the static price in the total price and/or display it in the other prices section.

ADMINISTRATION

Managing Attribute Value Pricing

Pricing for attribute values is defined along with the values when creating an attribute.

1. Navigate to a Single Select or multi-select Menu.
2. Select **Static Pricing** for the **Attribute Value Pricing** option in the Main Information section.



3. Do one of the following:
    - Create **new menu values** and add pricing for each available currency **OR**
    - Click **Entry Manage**r to add a price to individual menu values



When AVP price is not given for a certain currency, the price for that currency is calculated using the AVP price in the base currency and the defined exchange rate.

When AVP price is not given for a certain currency and exchanges rates are not defined, the user is shown the message that the price is not defined.

4. Set Attribute Value Pricing Options:



a. **Pricing Display Options:** Choose None if you don't want to display the attribute value price or choose Show Pricing Next to Attribute Values.

b. **Display Price at Bottom to User:** Select to display the attribute value price in the Price section on the configuration layout.

Depending on the configuration layout you've chosen, this might not be at the bottom.

c. **Include in Total Price:** Select to include the attribute value price in the Grand Total in the Price section on the configuration layout.



Zero static price is not shown in the Pricing Section - If the AVP price for a menu attribute is zero, then this price is not shown in the pricing section.

NOTES



Attributes displayed on the Search Flow and subsequently on the Configuration Flow exhibits AVP pricing behavior based on the value retained from the Search Flow - When an attribute is displayed on the Search Flow and then displayed on the configuration flow, the AVP pricing behavior of this attribute depends on the value that is carried from the Search Flow. The user cannot save changes to the value of this attribute.* It is recommended that the fullaccessuser either hide the attributes carried over from the search flow or make them read-only for the benefit of the end user.

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""



Negative prices are not supported.

RELATED TOPICS

ATTRIBUTE TIPS

OVERVIEW

Attributes

- Avoid overriding menu attributes at lower levels in the product hierarchy.
- Overriding of AVP attributes is not possible at lower level; Attributes cannot be overridden at the lower level to include static pricing. All AVP features must be set up at the level where the attribute was created.
- Use the Set Properties Tab to define array attributes and easily change association to other sets. This tab appears only when the configurable attribute is of the type: Array.
- Don't have displayed information be case sensitive or be a condition for a rule (i.e., Part Numbers).
- Have as few attributes "auto-update" as possible.
- The more Attributes that exist in a flow, the slower the performance.
- Define Attributes at the level in which they are applicable. If two entities in a level use the same Attribute, move the Attribute to the parent level; if the Attribute is used in multiple models, define it at the Product line.
- Limit the length of an Attribute variable name.

Arrays

- Single select attributes cannot be displayed as radio buttons for use in sets.
- An array attribute cannot be of multi-select menu type. If on the Attribute Editor page you choose the attribute type as a multi-select menu and then select the array type option, then the attribute will automatically change to text-field.
- Once you define an attribute as an array, you can't change it to a non-array attribute.
- You can create configurable attributes with the same label name, though the variable names must be unique. If your application is pre-12.2 and you need to create two or more attributes with the same label, the following may help:
    - “My Attribute” and “My Attribute” are rendered the same in the config UI, even though they will appear as distinct attributes on the admin side. You can insert as many spaces in between the words of a label, and these spaces will not be visible on the user side.
    - For one word attributes, you can insert a non-breaking space copied from a character map, or entered via the keyboard shortcut ALT+0160. Please note that typically only numberpad keys work for this shortcut sequence.

Image Menus

- The Image Menu option is not available for non-text menus.
- Image menu attributes can be used in Configuration Rules and BML. For example, the label can be used in all Simple Conditions while the variable name of the menu value is used to define BML rules. Rule messages will show menu values as text and not images.
- Bulk Upload and Download of Image Menus are supported.
- Menu values selected in configuration are carried to commerce as text. The image URL is not available in the sub-document and can't be rendered in the main document or the sub-document.
- Image Menus are not available as Dynamic Data in the Main Document Rich Text Editor. For sub-document RTE, Image Menus can be added as dynamic data to the template. However, if the image tag is chosen, then the URL will be printed as text and the image will not be rendered.
- Before you being with Image Menus, it is recommended that you update the Configuration Stylesheet. Simply navigate to All Product Families > Stylesheets and click Update.



Upload your images as JPG or non-transparent GIF format. PNG is not supported in Doc Engine.

NOTES



The order of configurable Attributes is the primary driver for firing rules.



Once created, Attribute types cannot be changed.

RELATED TOPICS

ARRAY BEST PRACTICES

OVERVIEW

Building Arrays

1. Configurable array sets must be created the proper way in order for the +/- add/remove buttons to appear or work .
    - Create the array set BEFORE visiting the “Set” tab of the array attribute
    - Ensure that the array control attribute is in the same group as the array set, and is not hidden.
    - This order will always work:
        a. Create the array control attribute
        b. Create the array set
        c. Create the array attributes and add to the set
2. If you created the wrong way you DO NOT need to delete your attribute and recreate. Here are the steps to fix an Array Set that is not showing the increment / decrement buttons:
    - Do the following for each array in the array set:
        a. Go to the Array Attribute on the admin side.
        b. Go to the Set Properties tab.
        c. Change the Array Set from the real one to the [DEFAULT] one.
        d. Change the Array Control Attribute to the blank entry.
        e. Click Apply. *important step*
        f. Change the Array Set back to the real one.
        g. Click Update.
    - Once you have done that for all arrays, the little icons should return.
3. It is possible to have one Size attribute control multiple arrays, but not recommended, since it will lead to confusion in maintenance.

Recommendation Rules and Arrays

- Set Type Recommendations will work for an array attribute only if none of the values of those attribute elements is changed by the user. Once changed, the system treats it as an overwrite of the attribute and will not perform a Set recommendation on any elements that get added afterward.

- If you recommend a value for a string array of "" (blank) or $BM_UNCHANGED_STR$ (leave value unchanged), even if the value is blank, you will still get the recommendation message ("Don't select any value…").

User Experience with Arrays

- There is quite a bit of BM behind-the-scenes Javascript associated with array sets. The presence of many array sets will affect performance, especially in IE.
- When you add a new element to an array using the +/- buttons, the value it takes is the value of the previous last element (the now penultimate element). Defaults are not respected in arrays.
- The recommendation text for an array attribute always shows the variable name of the attribute, rather than the display name, as for non-array attributes.
- Maximum number of rows in an array is 200.
- Multi-select menus are not possible in arrays.

Arrays & Table-based Rules

- Simple Table-based Rules operate per index; they will not operate across indexes.
- Attribute type limitations are consistent with existing simple Table-based Rules.
    - If the Table column is numeric, then the Array Attributes must be numeric

---

NOTES



Maximum Array size is 200.

RELATED TOPICS

GUIDED SELLING

OVERVIEW

A search flow, or selector, offers more granularity and flexibility in navigating the product hierarchy, prior to entering the configuration flow. The selector can guide users to the correct model when they do not know what they are looking to configure in advance. For example, this would be necessary if the users do not know the products or if there are complicated calculations required to select the model.

ADMINISTRATION

Step 1: Creating a Search Flow

Please click here for information on Creating a Search Flow.

Step 2: Set Up a One-Node Flow

1. Click **List** under the Rules header for your Search Flow.



Selectors can have multiple flow nodes, just like configuration. This is useful when the fields for the user change significantly based on some input. Recommendations, constraints, and hiding rules can also take effect in selectors.

2. Click here to see steps for creating a Search Flow Rule.

3. Select Define Function as the Result Type.



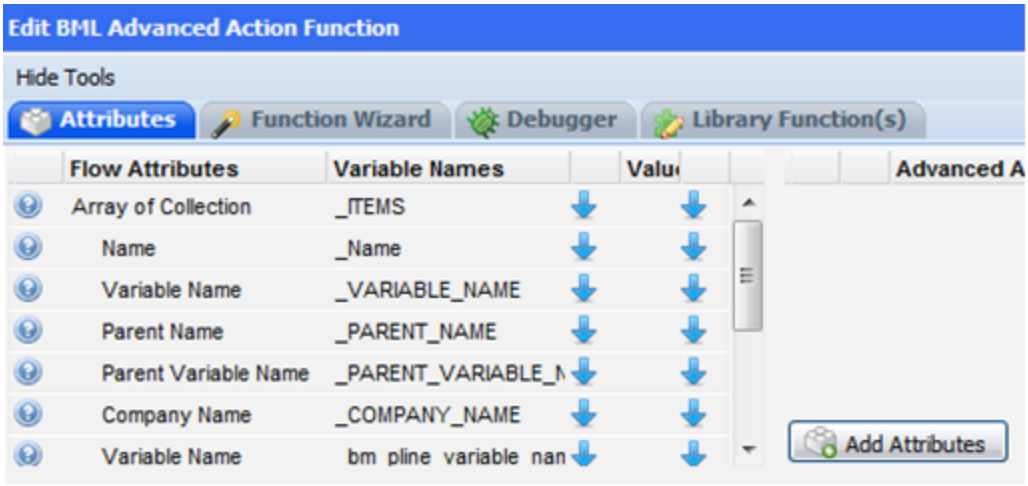
This script determines which model punch-ins will appear to the user, based on any condition attributes that you have added to the search flow.

4. In this example, Legacy Type and Product Line are single select menus. Their values (New, Old and X, S, NL) will determine which models appear, depending on whether the model is New or Old and whether it belongs to the X series, S series or NL series. *The BML script automatically imports the catalog attributes as parameters. Not all of these may be relevant for your use case.*

5. There are a few special attributes of the XML tags:

   - **button='yes':** For product line or model, show as a button. 'No' will show as a link.
   - **configuration='yes':** For model, destination of the button or link will go to configuration. If 'no', it will go to the model profile page.

- **configuration='yes'** :For model, destination of the button or link will go to configuration. If 'no', it will go to the model profile page.
- **sort='yes':** This attribute is sortable on the selector results page for the end user.
- **numeric= 'yes':** Define the sort behavior as numeric, so values are 1.0, 2.0 10.0 rather than string sort of 1.0, 10.0, 2.0.



6. The BML script returns a string containing XML. The XML has to conform to a specific hierarchy. Click here for an example.

```
<SelectorSearchResults>
    <Columns>
        <Column ... />
        <Column ... />
        ...
    </Columns>
    <Items>
        <Item>
            <Attributes>
                <Attribute ...>attributeText</Attribute>
                <Attribute ...>attributeText</Attribute>
                ...
            </Attributes>
            <Description>
                <GroupName hidden="yes" button="no">item._PARENT_VARIABLE_NAME</Grou
                <Name button="Yes" title="item._NAME" configuration="yes" align="cen
            </Description>
        </Item>
    </Items>
</SelectorSearchResults>
```

7. Click here to see the full script used in this example, with comments.

```
str = "<SelectorSearchResults>";

//Columns: These are the columns that show up in the selector search results
str = str +"<Columns>";
str = str +"<Column title=\"Product Line\" align=\"center\" width=\"25%\" sort=\"tru
str = str +"<Column title=\"Legacy Type\" align=\"center\" width=\"25%\" sort=\"true
str = str +"<Column title=\"Model\" align=\"center\" width=\"25%\" sort=\"true\">3</
str = str +"</Columns>";

modelDict = dict("string");

//A data table that maps the model names available for the search flow's search para
modelMap = bmql("select modelName from ModelMap where productLine = $productLine_sel
for model in modelMap {
    put(modelDict, get(model, "modelName"), get(model, "modelName"));
}

itemStr = "<Items>";
for item in _items { //_items is a list of all Models across the product lines for t
    if (containskey(modelDict, item._NAME)) { //If modelDict contains this specific
        itemStr= itemStr+ "<Item>";

        itemStr = itemStr + "<Attributes>";
        itemStr = itemStr + "<Attribute title=\"Product Line\" variableName=\"produc
        itemStr = itemStr + "<Attribute title=\"Model\" variableName=\"legacyType\"
        itemStr = itemStr + "</Attributes>";

        //This column needs to have the hidden GroupName node or the Model buttons w
        itemStr = itemStr + "<Description>";
        itemStr = itemStr + "<GroupName hidden=\"yes\" button=\"no\">"+item._PARENT_
        itemStr = itemStr + "<Name button=\"Yes\" title=\"" + item._NAME + "\" confi
        itemStr = itemStr + "</Description>";

        itemStr = itemStr + "</Item>";
    }
}
itemStr= itemStr +"</Items>";
if (itemStr <> "<Items></Items>") {
    str = str + itemStr;
}

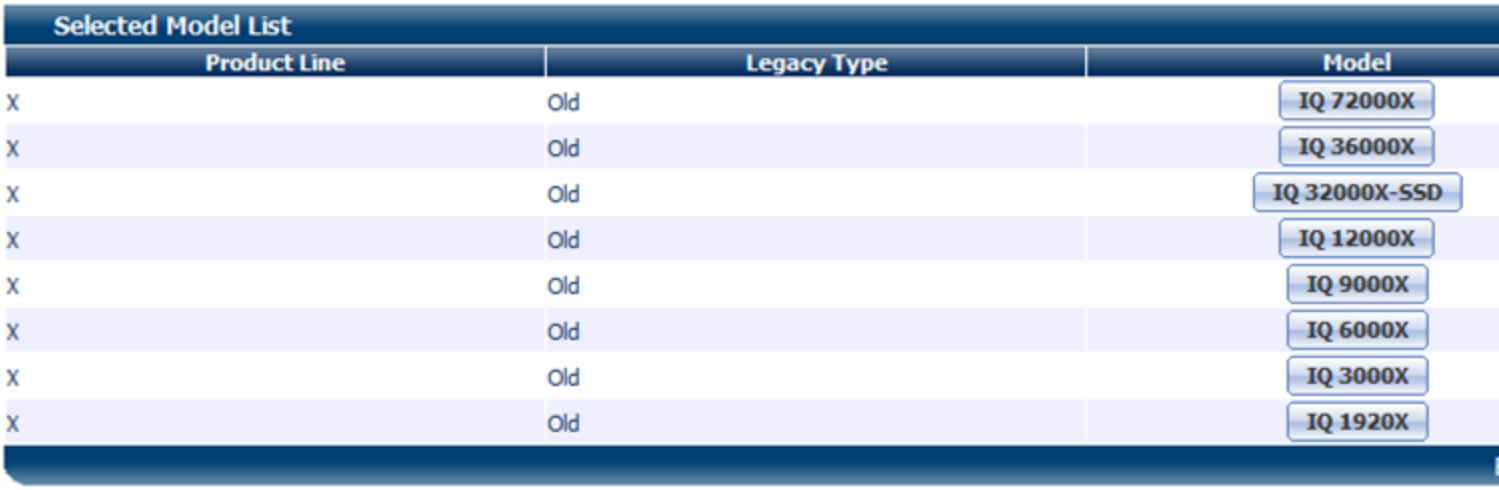
str= str +"</SelectorSearchResults>";

return(str);
```

8. Refresh and deploy the home page. Then, from the home page, enter the Selector. You will see a screen like this:



9. And entering the Search Result screen:



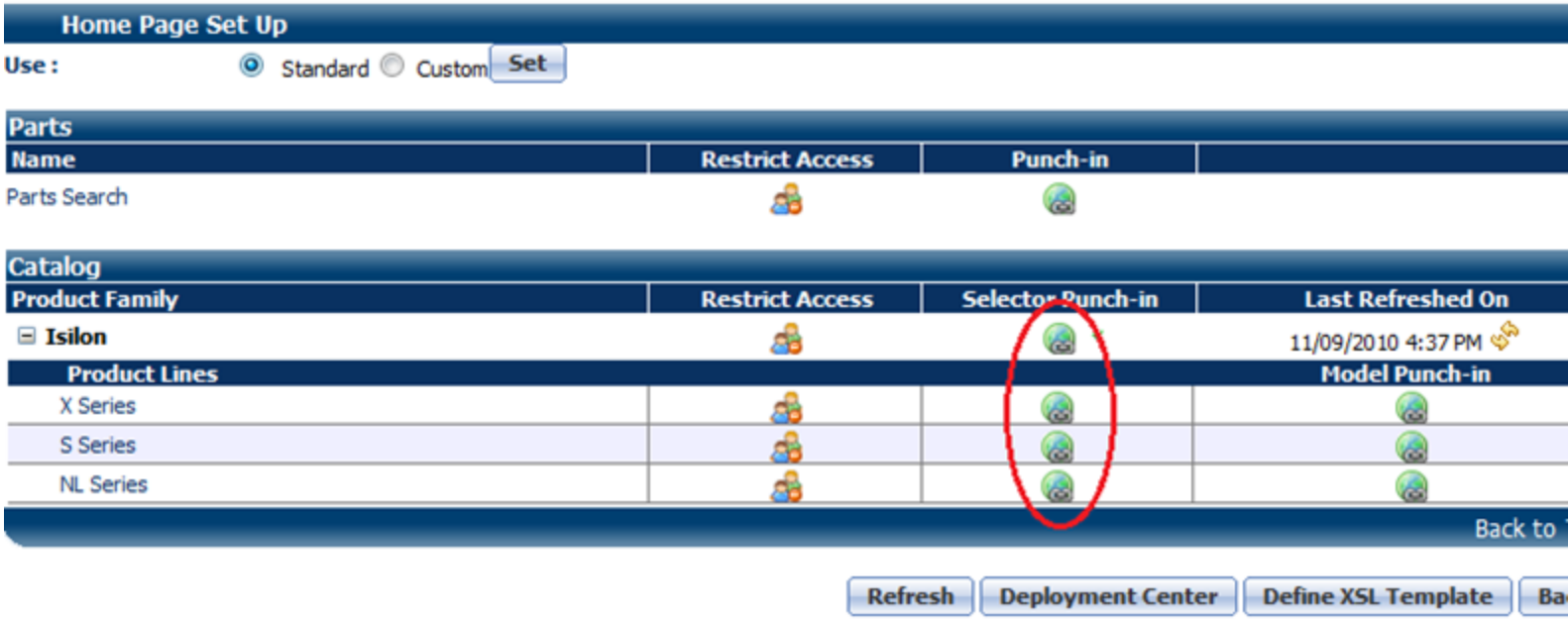


You may need to deploy regularly in configuration if the BMI instance you are working on has its configuration cache turned off.

Step 3: Changing the Homepage UI and the Models that Your Selector Can Access

When you create or modify your search flow, choosing its Type affects the corresponding layout on the Homepage to a certain degree. In most cases, you will probably choose "Choose models across product lines". Upon selecting this option and deploying the Homepage, a selector will be generated for each product line. So if you have 7 product lines, there will be 7 punch-ins to 7 selectors, each of which is only capable of accessing the models that belong to that specific product line. This is probably not what you had in mind. Additionally, you are probably averse to having what looks like 7 identical punch-ins cluttering the Homepage.

By default, a selector punch-in is generated on the Homepage only for each product line. So, you will need to go to Homepage admin and create a selector punch-in for the entire product family.



Note the items circled in red. There is a selector punch-in for each product line in the product family, and by default this is probably what gets generated when you first refresh and deploy the home page. In order to create a selector that can access all models across all product lines, you need to create a selector punch-in for the Product Family. In this example, you will have to click on the selector punch-in icon that corresponds to "Isilon". This will bring you to the selector punch-in list for the Isilon product family:

In this example, a punch-in for the product family has already been created. In your case, you will probably need to create a new one. Name it anything you like.

You may run into an error message that pops up saying something similar to "search group not available for this selector". A work-around for this is to go back to the Search Flow and change its type to "Select product lines". You will have to unselect all product lines. Then, deploy config. Go back to Homepage admin, create a selector the entire product family (this should now work). Once that is done and the homepage is deployed, go back to the search flow and change its Type back to what it was before (don't forget to select the product lines again).

RELATED TOPICS

CONFIGURATION RULE OVERVIEW

OVERVIEW

Rules enable you to set specific configuration information for your products. For example, you can suggest configured values to buyers using recommendation rules or hide some configurable attributes from certain user types. Rules can also be used to recommend items to buyers, prevent them from selecting opposite values of configurable attributes and display additional price information.

Types of Configuration Rules

Rule Type	Description	Example
Bill of Material (BOM)	Builds a BOM string based on configuration.	When a user selects crust type = deep dish, a BOM string with sauce, cheese, butter and type of dough is generated and returned to an ERP system.
Pricing	Calculates a price based on how a product is configured.	When a user configures a model, the price is displayed based on the crust type, pizza size and topping selected.
Hiding	Hides attributes when a pre-defined condition is met.	When a user selects pizza type = specialty, the toppings attribute is hidden.
Recommendations	Sets items you would *like* the users to select OR items the users *must* select.	When a user selects specialty pizza = Meat Lovers, the application recommends only pepperoni and sausage as toppings.
Constraints	Determines attributes values users cannot select within a configuration.	When a user selects pizza size = small, they cannot select Deep Dish as the crust type.
Recommended Items	Assigns parts and pricing to the configuration, based on certain selections.	When a user selects sandwich type = Hot, the application will return 4 recommended sandwich options.



Rules should always be created at the highest level possible in the hierarchy, though they are available for creation at any level.

Common Rule elements

- Rules are created individually, at any level in the system. (Except for search flows, that can only be created at a product line level)
- Rules can be made active or inactive
- Rules abide by the principles of inheritance, so rules created at a high level are automatically inherited by lower levels. Usually, rules created at a lower level are given precedence over rules created at a higher level.
- When more than one of the same rule type can be triggered at the same time, the configuration system evaluates which rule to trigger based on rules precedence.
- Rules evaluate a certain condition and perform some action.

- The values of the attributes selected as the condition attribute(s) determines the results of the condition. If the condition is evaluated as true, then the action is performed. The action may involved action attributes. The condition take in Configurable Attributes, Regular Attributes, User Attributes and Account Attributes. The action attribute are Configurable Attributes.

Order of Operations: Rule Precedence

1. Config Flow Rules
2. Constraint Rules
3. Recommendation Rules
4. Constraint Rules (again)
5. Hiding Rules
6. Recommended Item Rules
7. BOM Rules



For more information on how reordering attributes affect rule precedence, see Configuration Attributes Overview.

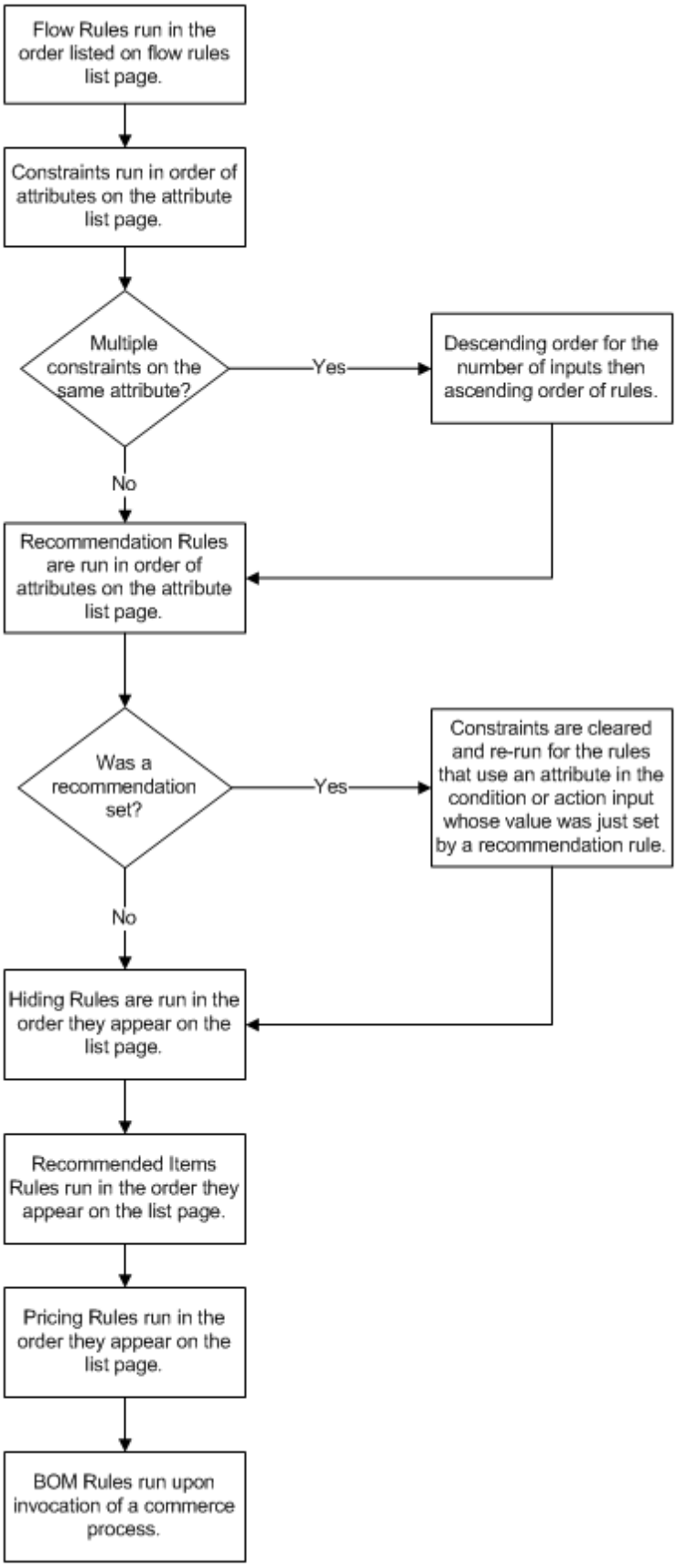


Rules are executed in the order that their related Configurable Attribute appear within in the Configurable Attributes Administration List. *In general, most users will not need to reorder their attributes; only a few special circumstances warrant attribute reordering.*

Configuration hierarchy

The data entered on the configuration page will be read before any rules are run. The order of configuration hierarchy inside of each of the nodes is:

1. Model
2. Product Line
3. Product Family
4. All Product Families


Flow Rules run in the order listed on flow rules list page.
Constraints run in order of attributes on the attribute list page.
Multiple constraints on the same attribute?
Yes
Descending order for the number of inputs then ascending order of rules.
No
Recommendation Rules are run in order of attributes on the attribute list page.
Was a recommendation set?
Yes
Constraints are cleared and re-run for the rules that use an attribute in the condition or action input whose value was just set by a recommendation rule.
No
Hiding Rules are run in the order they appear on the list page.
Recommended Items Rules run in the order they appear on the list page.
Pricing Rules run in the order they appear on the list page.
BOM Rules run upon invocation of a commerce process.

ADMINISTRATION

Editing a Rule

1. Navigate to the Rules List page for the required rule type at the appropriate level in the product hierarchy.
2. Click the desired rule name in the Rules List to launch the Rule Editor.
3. Use the Rule Editor to change the Name, Description, Status and Condition Reference Type. You can also re-select condition and/or action attributes, specify different values for the condition attributes, and edit actions.

Deleting a Rule

1. Navigate to the Rules List page for the appropriate rule type at the correct level in the product hierarchy.
2. Select the rule that is to be deleted.
3. Click on Delete.

Reordering Rules

1. Navigate to the Rules List page of the appropriate rule type at the desired level in the product hierarchy.
2. Enter the desired order number in the Order column of the rule.
3. Click on Update. This will reorder the rules list. The rule that occupied the selected position before the reordering will now appear in the ertswhile position of the selected rule.

NOTES



After creating rules, you can test their usage with a tool called the Pipeline Viewer.

If you system has configuration cache enabled, you must deploy configuration in order to update the cache and enable your changes.



If you want to create and frequently test configuration rules without having to deploy the changes, you can contact your System fullaccessuser to temporarily disable the configuration cache.

RELATED TOPICS

GENERAL RULE ELEMENTS

OVERVIEW

A configuration rule is comprised of a condition and an action. The condition is always evaluated first. if the condition is evaluated to be true (meaning it is met), then the rule is fired (meaning an action will take place).

Each rule begins with a basic template of 3 sections: Properties, Condition, and Action.

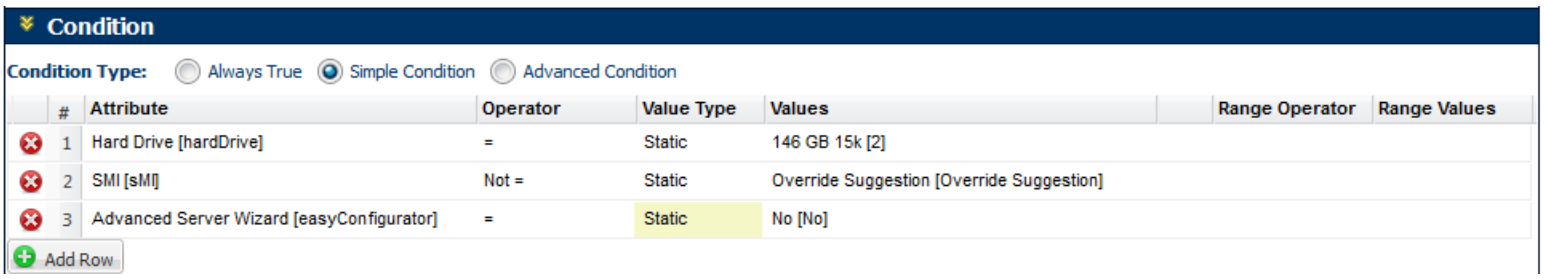
Properties

This section is where you will enter basic details about the rule. For example, the name, status and where it will be applied.



Label	Description
Name	Enter the name of the rule. **This is required.**
Variable Name	Enter a variable name for the rule, but keep in mind it cannot be changed. **This is required.**
Description	Enter a description, if desired.
Status	Determine the status of the rule: • Active: Rule is fired when the condition is met. • Internal: Rule is active only for FullAccessUsers; it is inactive for all other users. • Inactive: Rule is not fired, even if the condition is met.
Edit Start/End Dates	Select the box to open a date selector to set how long the rule will remain in the chosen status. If no duration is selected, the rule will remain on the selected status until you, or another fullaccessuser, changes it.
AJAX Enabled	Upon checking this box, you are enabling AJAX functionality for the rule. This means that the rule will run without refreshing the page or requiring an action be performed. Please note: AJAX functionality is ONLY AVAILABLE for Hiding and Constraint rules.

Condition



The condition section allows you to select the logic that specifies when a rule should run. There are three options: Always True, Simple Condition and Advanced Condition.

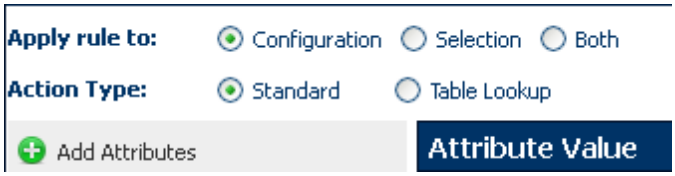
- **Always True:** The rule will fire automatically, every time, because a specific condition does not need to be met.
- **Simple Condition:** The rule will fire based on define condition attribute values that you select from a menu.
- **Advanced Condition:** The rule will fire based on an advanced function written with BML.



Conditions are returned as Boolean.

Action

The action section is where you select WHAT will happen if the condition is met. This is also where you will select where you'd like to apply the rule to; Configuration, Selection or Both. If you are creating a Recommendation or Constraint rule, this is also where you will select if you'd like to select action attributes by clicking the Add Attributes button or by a Table Lookup.



Rule Type	Action Description
Bill of Materials (BOM)	No action attributes are defined. The action will be a BOM string that is returned to an ERP.
Pricing	No action attributes are defined. The action will be the price that is returned.
Hiding	Define which attributes will be hidden.
Recommendation	Define which attribute(s) will have it's value(s) set or modified.
Constraint	Define which attribute(s) will have value(s) constrained or allowed.
Recommended Items	No action attributes defined. The action will return mandatory or optional part numbers and quantities.

NOTES



BOM, pricing rules, and recommended item rules do not require the users to choose Action Attributes. For these rules, the users must specify the action only.Combine configuration rules as much as possible. Separate rules are harder to maintain.

When relevant, add comments to explain why specific design decisions were made.

Keep rules on the same level, making them easier to maintain and debug.



Rules can only be deleted at the level of the hierarchy in which they were created.



Deleted Rules cannot be recovered.

RELATED TOPICS

RECOMMENDATION RULES

OVERVIEW

What are they?

Recommendation Rules can be used to help buyers configure your products by offering suggested attribute values. For each configuration where a model (or part) would likely have certain attribute values, you can create a recommendation. For example, a Meat Lovers pizza has only pepperoni and sausage as toppings and this can't be modified by the user. A recommendation rule can achieve this.

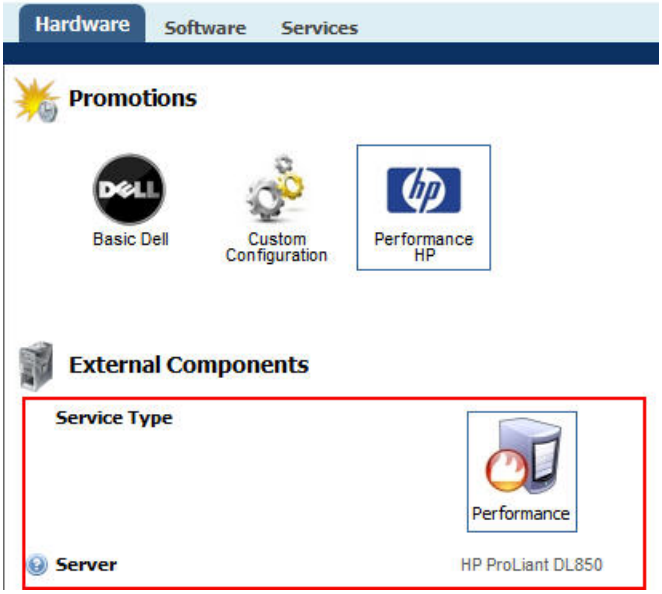
How do they work?

When recommendations trigger, they appear to buyers on the Model Configuration page. By default, recommendations display as text next to the configurable attributes with recommended values. They also use set-types, that when enabled, can auto-populate the recommended values in the attribute fields.

Set Types

Set Type	Description
None	The selected value for the action attribute in the rule will be recommended value for the attribute. It will show as a message and it can be modified by the user.
Set	The setting of a recommended value can be prevented by locking the attribute, in which case, only the recommended values message is shown. This can be modified by the user.
Forced Set	The rule will set the attribute value to the value chosen in the rule. This value is read-only and cannot be modified by the user.

If you look at the image below, a recommendation rule is running that makes Service Type and Server read-only.



Multiple Recommendations firing simultaneously on an Action Attribute

When there are multiple recommendations firing simultaneously for the same action attribute, then the rules are first sorted according to the following criteria (in sequence):

- Descending order by product hierarchy (model-level first and all-equipment level last)
- Descending order by number of rule inputs (rules with single inputs are last)
- Ascending order by the order number. See Reordering Rules.

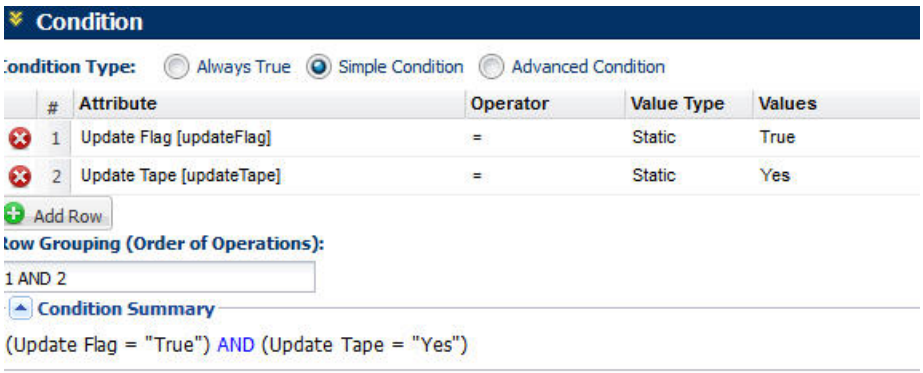
Recommendation Rules and Search Flows

Recommendation Rules are applicable in Search Flows. If a Recommendation Rule is defined at the Product Family level, then you are given an option to specify the area of application of the rule. See Editing Recommendation Rules.

ADMINISTRATION

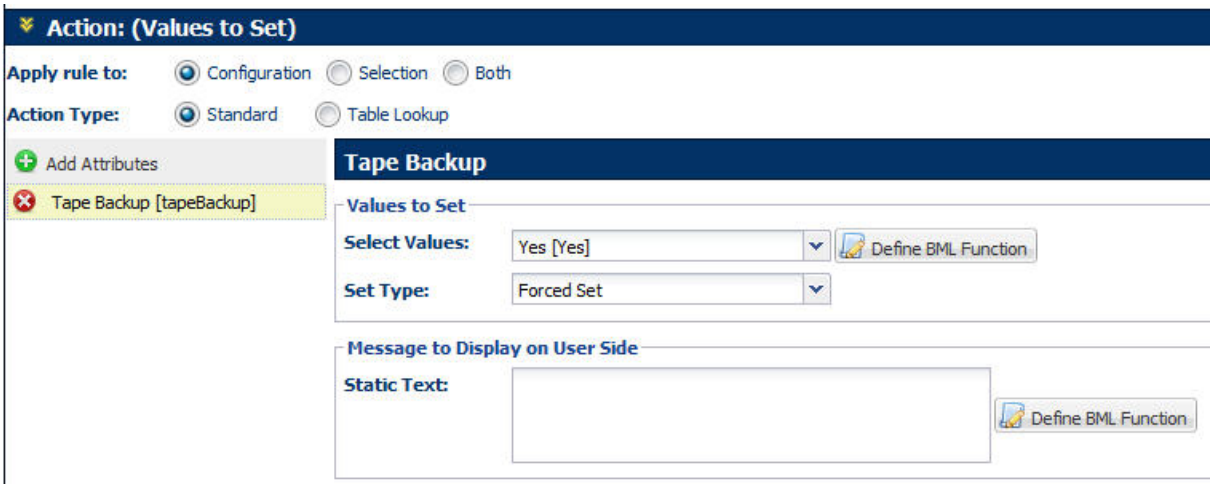
Managing Standard Actions

1. Navigate from the Admin Home Page > Product Family > Recommendations > List
2. Enter the **Properties** of the rule.
3. Select your **Condition Type**.



4. Choose to apply the rule to either **Configuration, Selection or both**. *Configuration is the default selection.*
5. Choose the Action type:
   - **Standard:** You will use the UI to add attributes.
   - **Table Lookup**: Select attribute values from a table. See Table Based Rules for more information.
6. Choose an attribute whose value needs to be set. Under Values to Set:
   1. Choose attribute values from the **Select Values** menu, OR
   2. Click **Define BML Function** to set values through a BML script.\

7. Choose the **Set Type**:
    - **None:** The user can input any value without encountering an error message.
    - **Set:** A recommendation rule can change the value entered by the user unless the attribute is locked by the user.
    - **Forced Set:** A recommendation rule will change the value of the attributes regardless of what the user enters and if the attribute is locked or not.



8. Enter text into the Message to Display on User Side text box. Enter a Static Text message to the user by entering text into the

> Click the Define BML Button and write a script that returns a string. This string is displayed as the comment. See Pricing Rules for an example.

9. Click **Update and Back** to save and return to the Recommendation Rule List page.

NOTES

> Recommending a number that has more than 6 decimals is not recommended. Functionality will not be affected, but it will generate an error in the admin error log.

RELATED TOPICS

RECOMMENDED ITEMS

OVERVIEW

What are they?

Recommended item rules enable you to associate parts and models with products based on user-configured values. The items recommended may be models or parts. If the recommended item is mandatory, then the user must buy the model selected with the recommended item. There is no way to delete the item association in the commerce process and buy the selected model without the recommended item. If the item is not mandatory, then the user can opt to not buy the recommended item.

How do they work?

A Recommended Item rule also has a condition and an action. The values of the attribute(s) selected as the condition attribute(s) determine the result of the condition, which when true triggers an action to show the recommended items models and/or parts selected.

There is a site option to allow the display of the price of the recommended items on the model configuration page. Also, the comment for the rule can be a simple comment or can be specified through an advanced rule.

Recommended Items Pricing

You can define prices for your recommended items, other than the base price of parts or models. When you define prices for the recommended items in your Recommended Items Rules page, this price is given highest precedence over the other prices defined (price book and/or base price). If you do not define any recommended item advanced pricing rules, the system looks for the prices defined in the price book and displays this price to the user. In case the price book is turned off, then the system displays the base price for the part or model to the user.

When several rules recommending the same item are fired at a time, then the recommended item price is picked up from the rule that is last in the order. The quantity of the recommended item is decided by adding the quantities that each of the rules provide. This is not true if the option to not sum is turned on.

How Recommended Items Pricing Works

If you define the recommended items advanced pricing, then the price of the item is given by the price specified in the rule. If you do not specify price in your rule:

- For a model, the base price is passed to the commerce transaction.
- For a part, if price book is available and the user has access to it, then the price book price is passed to commerce.
- For a part, if a user has access to multiple price books, then the user must choose the price book on the model configuration/recommended items page. The price from the chosen price book is then passed to commerce.
- For a part, if price book is not available, then the price is given by the base price of the part.

ADMINISTRATION

Creating a Recommended Item Rule

1. Navigate from Admin Home Page > Catalog Definition > Product Family > Recommended Items > List.
2. Enter the **Properties** of the rule.

3. Select your **Condition Type**.
4. Choose to apply the rule to either **Configuration, Selection or both**. *Configuration is the default selection.*
5. Select **Mandatory**, if the recommended items are required and should display as read-only.
6. Type any **Comments** that you would like to display with the Recommended Items.
7. Choose the **Action** type:
    - **Advanced:** You will use a BML script to return models or parts to the rule.

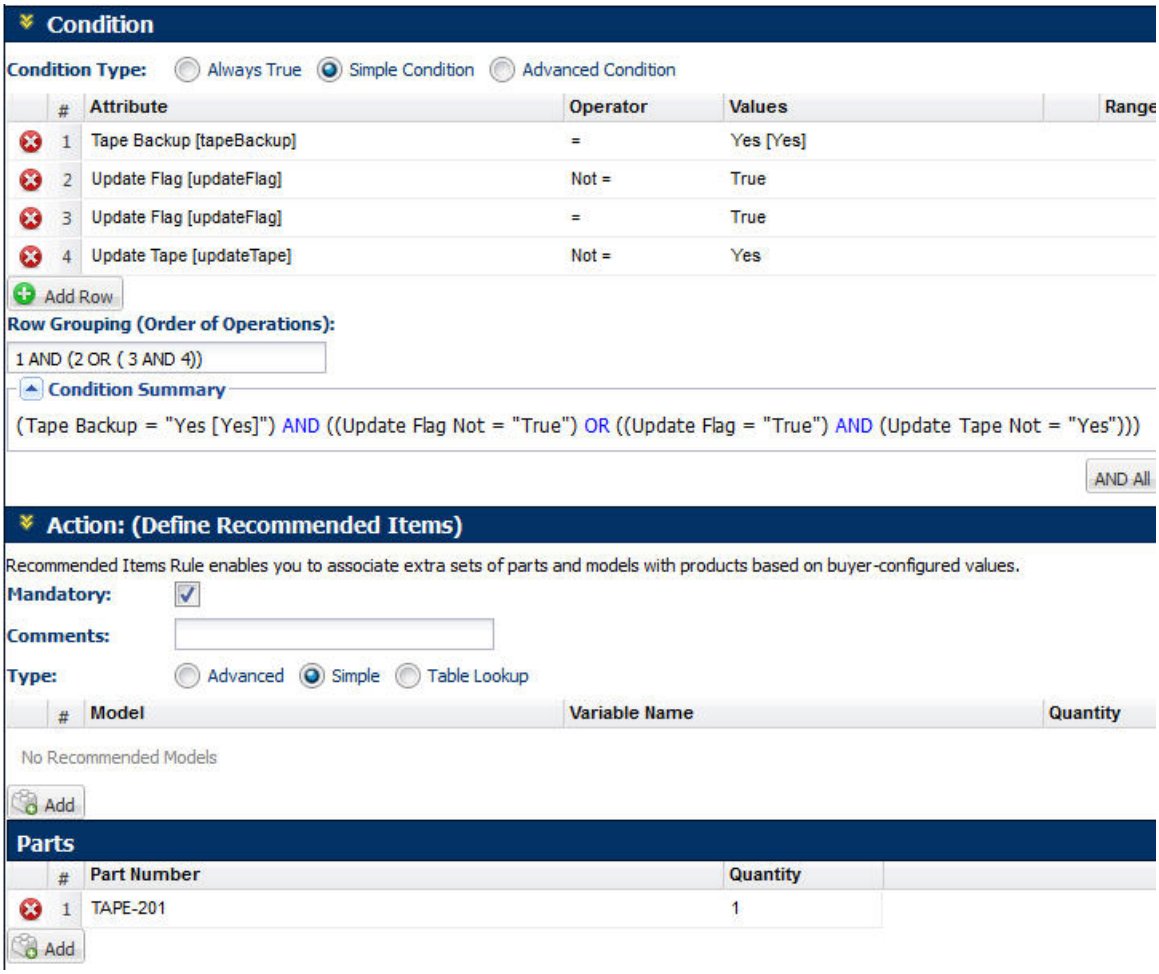
      

      The expected return format is (item1~quantity1~comment1~price1[~identifier]|^|item2~quantity2~comment2~price2[~identifier2])

      Identifier is an optional parameter that is used to define a unique identifier. When part numbers are grouped, they will only be grouped if they have the same identifier.

      In this format, price is not mandatory in the string format. If you pass 0.00, the users get ”r;On Request” against the price field on the Model Configuration/Recommended Items page. If you do not pass a price or pass an invalid or a blank string, the price of the item is picked up from the price book/base price.

    - **Simple:** You will use the UI to add parts or models to the rule.
    - **Table Lookup**: Select attribute values from a table. See Table Based Rules for more information.
8. Add the part or model and the quantity of each that you would like to recommend when the condition is met. In this case, it is a part. Click **Add**.



9. Click **Update and Back** to save the rule and return to the Recommended Item Rules list page.

Recommended Item Splitting

In previous versions of BigMachines, multiples of the same part number, within the same rule, would show up as a summed quantity in commerce. If you wanted to see them broken out individually at the bottom of the page, you'd have to create different rules.

So let's say that you are using a recommended item rule to go through all of your indices of an array set:

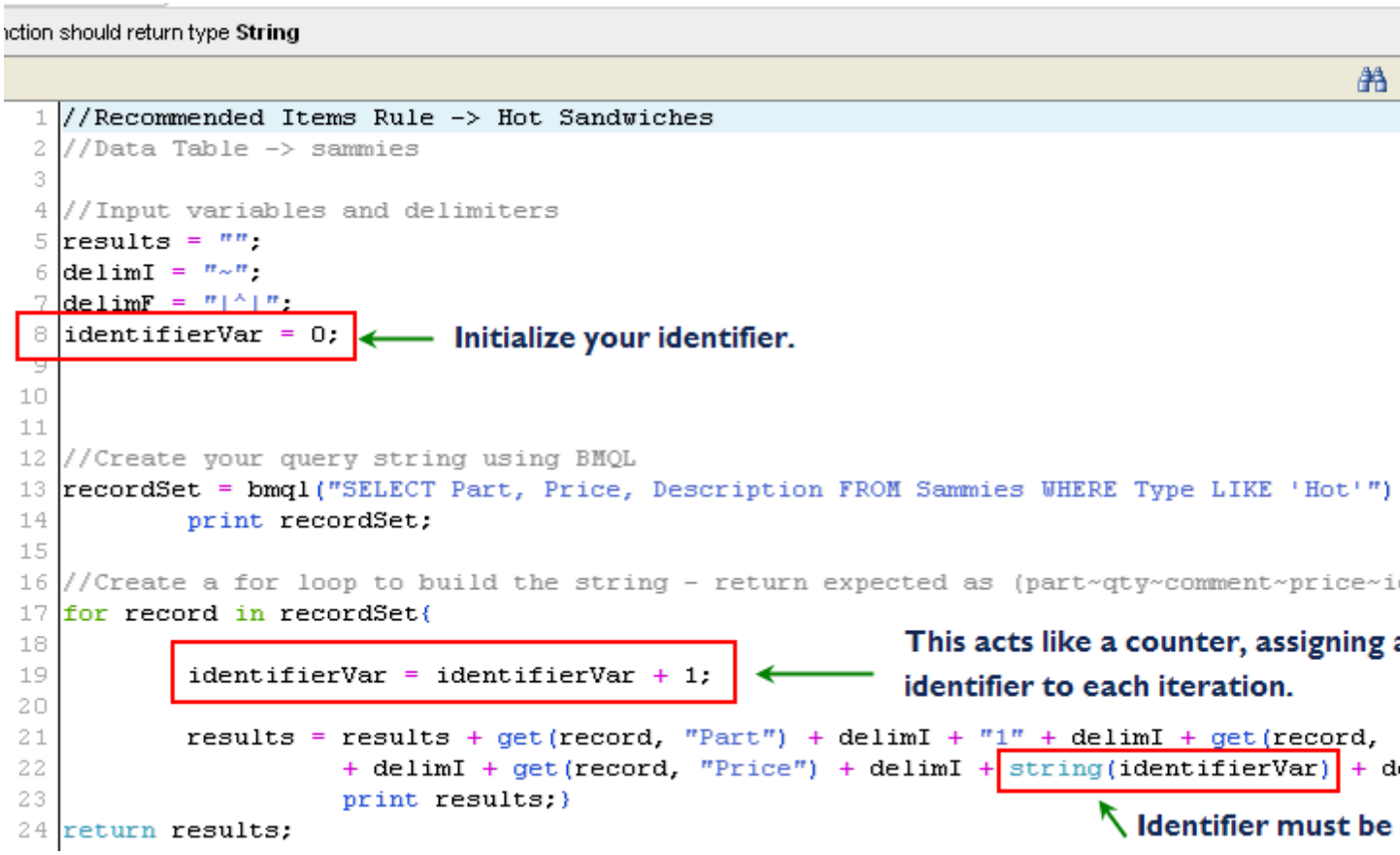
1	000S1	1
2	000P1	4
3	000S1	6

In previous versions, lines one and three would roll into one, but by selecting the Split Item option, you can now have these remain separate.

Sample Use Case

Scenario: You have Hot Sandwiches on your menu, but you want to ensure that the individual part numbers are split on the Recommended Items UI. You have stored the sandwich name, description, type and price in a data table called "Sammies". Write a rule to recommend all hot sandwiches in your data table.

1. Follow **Steps 1-6** in Creating a Recommendation Rule.
2. Select **Advanced** for the Action Type in Step 7.
3. Choose the option **With Advanced Price and Comments**.
4. Click the **Define BML Function** button to open a BML Editor.
5. Write a **BML rule using BMQL** and your data table. Please don't forget to comment your code. Please see the article BigMachines Query Language (BMQL) for more information.



6. Click **Save and Close** to return to the Recommended Item Rule List page.

   Now, on the user-side, when someone selects Hot Sandwiches, only hot sandwich types found in the data table will display as recommended items. Notice the items have been split.

Recommended Parts

Select	Part Number	Description	Price	Quantity	Comment
☑	HS001	Meatball Sandwich	$7.95	1	Meatball Sar
☑	HS002	Grilled Chicken Sandwich	$8.95	1	Grilled Chick
☑	HS003	Italian Beef	$6.95	1	Italian Beef
☑	HS004	Grilled Veggie Sandwich	$4.95	1	Grilled Veggi
☑	HS001	Meatball Sandwich	$7.95	1	Meatball Sar
☑	HS002	Grilled Chicken Sandwich	$8.95	1	Grilled Chick
☑	HS003	Italian Beef	$6.95	1	Italian Beef
☑	HS004	Grilled Veggie Sandwich	$4.95	1	Grilled Veggi



This advanced use case can also be accomplished by selecting the Split Items option and Table Based Rules for the Action Type.



This functionality is not available for use in commerce.

This is not a way to sort line items before moving into commerce. Currently, sorting functionality is not available.

The default identifier is the variable name. This allows items to be bundled across rules. Items with the same identifier across rules will roll up.

You can use the identifier to sum items.

Sum Recommended Item Quantities

An option, Sum Recommended Items Quantities, is available in Configuration Settings. By Default this is set to yes. If this is set to no, the quantities for the same item that is recommended by different rules will not be summed and these items will appear as separate line items in the commerce document. Quantities are still summed if the same item is recommended multiple times by the same rule or by different rules with the same variable name. Please click here for more information.

NOTES



Mandatory items cannot be unchecked and will always be part of the main model at the time of invocation.

The user will not be able to invoke a transaction or reconfigure with missing/unavailable mandatory items.

Price defined for the recommended items are assumed to be in the base currency format.



The user will not be able to invoke a transaction or reconfigure with missing/unavailable mandatory items.

RELATED TOPICS

CONSTRAINT RULES

OVERVIEW

What Are They?

Constraint rules are used to let the user know which values are and are not allowed to be selected in a particular configuration. While a constraint is active, the user can not advance to commerce.

How Do They Work?

A constraint rule has a condition and an action. You can choose attribute values for the condition, which will when true, will trigger an action to be performed. Similarly, you can select action attribute(s) values that can be constrained or allowed. You can also enter a message that alerts the user that the configuration cannot have the selected set-up. The rule becomes inactive when the buyer re-selects values that are appropriate for either the action or condition attribute(s). Constraint actions can be defined as simple-action or table-based actions.

Multiple Constraint Rules on the Same Action Attribute

In the case of multiple constraints defined on the same action attribute, the rules are first ordered using the following criteria (in sequence):

1. Descending order by product hierarchy (model-level first and all-product family level last)
2. Descending order by number of rule inputs (rules with single-inputs fire first)
3. Ascending order by order number

The sorted rules are then applied on after the other to display the constraint messages in sequence.

Constraints and Recommendations

Constraints and recommendations can be applied together.

Constraints are applied to all the attributes, followed by recommendations and then constraints will be run again. Note: The constraints are applied in 2 passes because, the first pass of application of constraints determines the constrained action attributes to which the application of recommendations (if any) is skipped.

The recommendations are applied to only non-constrained action attributes. Constraints are run a second time to determine if any new constraints may have been triggered as a result of running the recommendations. So when the end-user makes changes to the attributes which are constrained and updates the configuration, new recommendations which were not run earlier (due to the earlier constraints) will also run which might trigger new constraints which are caught by the second pass of constraint application.

AJAX Constraint Rules

Constraint rules can run through AJAX resulting in faster performance and a dynamic user experience in configuration.

AJAX rules can be enabled for any constraint or hiding configuration rule. Unlike other rules, that are evaluated only if the attributes are set to auto-update or if a user clicks the Update action, AJAX rules are evaluated and the action is performed instantly when a user changes any rule input. Input can be condition and action attributes or action inputs. AJAX rules are re-run when a user clicks the Update button.

Constraint rule functionality will remain the same with AJAX. AJAX rules will support advanced condition and actions, table-based rules, and reverse constraint behavior. The exception to this are rules that have arrays as condition/action attributes or action inputs. AJAX constraint rules do not automatically hide the Pricing and Recommended Item sections when the user selects a constrained value. If the user performs an Update action, then these additional sections will be hidden. Also note that constraint messages that are displayed below the main content when constraints exist in some other tab are not removed via AJAX.

Recommendation messages will appear/disappear via AJAX for attributes that are inputs/action attributes of AJAX rules. It should be noted that this does not imply that recommendation rules are fired by AJAX. Recommendation rules will only be fired when the user performs an "Update" action. However, if a recommendation rule has already fired on an attribute and that attribute is a part of an AJAX rule, then the recommendation value setting and messages will occur via AJAX.

End-User Interaction

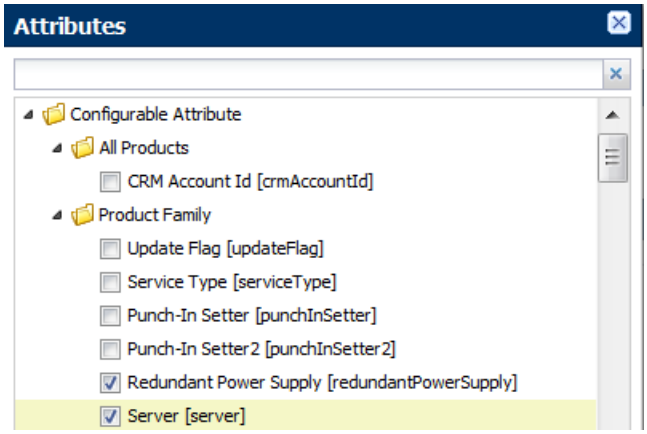
With AJAX rules, the end user will experience a faster, more dynamic model configuration page.

AVP (Attribute Value Pricing) will not automatically update is the user selects a constrained value that is constrained through an AJAX rule. Users will see the correct pricing upon performing an Update action.

ADMINISTRATION

Creating a Constraint Rule

1. Navigate to your Product Family > Constraints > List
2. Click **Add** to create a new rule.
3. Enter the **Properties** of the rule.
4. Select the **rule conditions**:
5. Choose to apply the rule to either **Configuration, Selection or both**. *Configuration is the default selection.*
6. Choose the **action type**:
    a. **Standard:** You will use the UI to add attributes to constrain or allow.
    b. **Table Lookup:** Select values to constrain or allow from a data table. See Table Based Rules for more information
7. Click **Add Attributes** to open the Action Attribute window.
8. Choose **configurable attributes** and click **Add**.



9. Select one of the following options:
    a. **Constrain All** to constrain all attribute values. Otherwise, leave it deselected and:
    b. Constrain Values:
        - If the attribute type is Text Field, the users can enter all the constrained values for the action attribute. If the data type of the attribute is either float or integer, then the users can specify a range. The action takes place if the condition evaluates to true and if the users enter any of the constrained values specified for the action attribute.
        - If the attribute type is a menu attribute, the users can select the constrained value(s) for the attribute. The attribute and its values appear in the Value column. The action takes place if the condition evaluates to true and if the users select any of the constrained value(s) specified for the action attribute. REVERSE CONSTRAINT: New for 10.1, the user can also return selectable items in a constraint rule by selecting "Allow Selected Values".
        - Non-menu values will not see the Constraint Behavior section as they are not affected and will remain the same.

    ---

    c. Define BML Function button to select attributes using a BML script.
        - If the attribute type is Text Field for both array and non-array attributes, (whether the display type is text or menu type) the function returns a "|^|" (pipe caret pipe) delimited string of all the constrained values for the action attribute. The action takes place if the condition evaluates to true and if the users enter any of the constrained values specified for the action attribute.
        - If the attribute type is Integer or Float, multiple constraints cannot be set using an advanced function. Only one constraint value can be set using the advanced function.

    ---

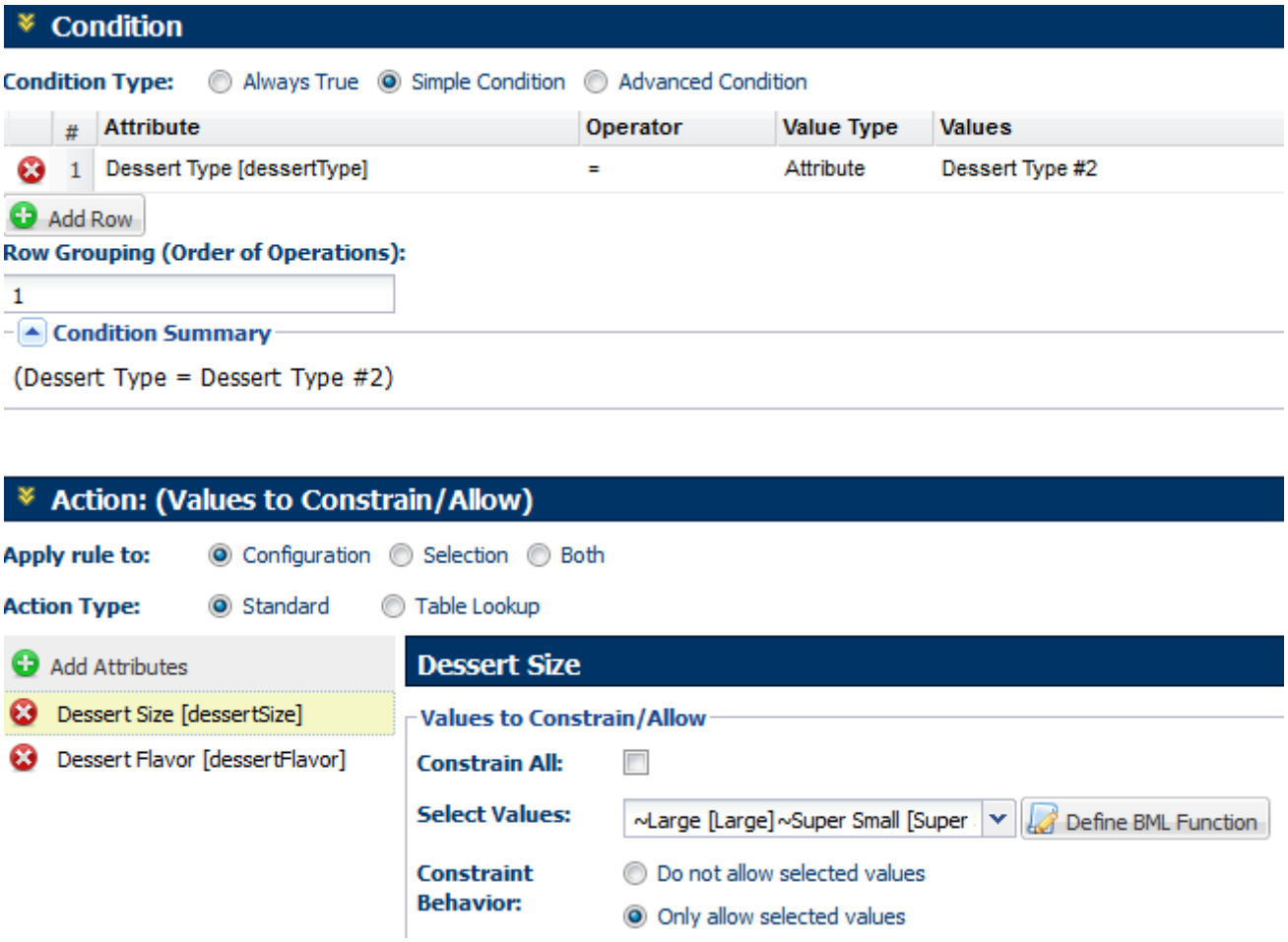
10. Choose the **Constraint Behavior**:
    a. Do not allow selected values (constrain)
    b. Only allow selected values (allow or reverse constraint)

11. Type a message in the **Static Text** box or click the **Define BML Function** button to create a dynamic message.



This message displays to the users only when the rule is run.

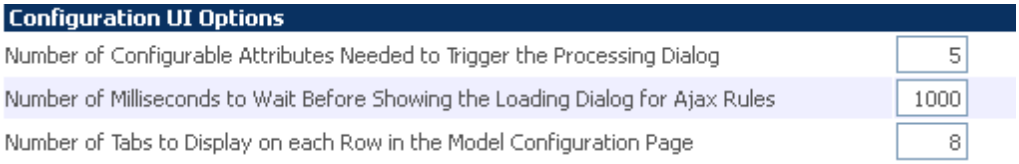
12. Click **Add and Back** to save and return to the Constraint Rule list page.



Enable AJAX

1. Navigate to the **properties** section of the rule.
2. Select the **AJAX Enabled** box.
3. Determine if you'd like to show or hide the processing dialog box for AJAX rules. *This is independence of the processing dialog box for the update action.*

1. Navigate from Admin Home Page > Products > Configuration Settings.
2. Enter a value for **Number of Milliseconds to Wait Before Showing the Loading Dialog for AJAX Rules**:
    1. **A positive number**: Sets how long the dialog wil appear (in milliseconds).
    2. **A negative number**: Disables the processing dialog for AJAX rules.
    3. **Zero (0):** Dialog box will appear instantaneously.



NOTES

AJAX Rules are supported on Search Flow and Model Configuration pages.

AJAX Rule with fewer action attributes - AJAX rules update the page using JavaScript post page loads. Hence it is important that we limit the number of attributes being affected per AJAX call.

If an AJAX rule fails to evaluate properly, the action will be timed out for 2 minutes. The end user can continue to interact with the page after that.

AJAX Rule without SOAP/ Table Queries –AJAX rules are not recommended to be used on rules with a large amount of BML, SOAP calls, or table queries as it could adversely impact performance and hence, end user experience.

AJAX rules are not supported for rules that have arrays as condition/action attributes or action inputs.

AJAX rules are not supported for the default wizard.

RELATED TOPICS

HIDING RULES

OVERVIEW

What are they?

Hiding attribute rules tell the system to hide select attributes when a pre-defined condition is satisfied. Like other rules, hidden attribute rules are made up of a condition and an action. The values of the attribute(s) selected as the condition attribute(s) determines the result of the condition, which when true triggers the hiding of the action attributes.

How do they work?

Using hidden attribute rules, you can reduce the number of flow rules needed for a configuration process. You can include different attribute types in a single flow rule and show one set of attributes in another, based on the same condition.

AJAX Hiding Rules

AJAX rules AJAX can be enabled for any constraint or hiding configuration rule. Unlike other rules, that are evaluated only if the attributes are set to auto-update or if a user clicks the Update action, AJAX rules are evaluated and the action is performed instantly when a user changes any rule input. Input can be condition and action attributes or action inputs. AJAX rules are re-run when a user clicks the Update button.

AJAX hiding rules do not support hiding the tab when all attributes are hidden in it. Similarly, a tab that is hidden will not appear via AJAX if the user “unfires” the associated hiding rule. The user must perform an Update action to view/hide tabs. Also, if a required attribute is hidden through an AJAX rule, no validation error will be thrown on Update or invoke actions.



Recommendation messages will appear/disappear via AJAX for attributes that are inputs/action attributes of AJAX rules. It should be noted that this does not imply that recommendation rules are fired by AJAX. Recommendation rules will only be fired when the user performs an “Update” action. However, if a recommendation rule has already fired on an attribute and that attribute is a part of an AJAX rule, then the recommendation value setting and messages will occur via AJAX.

Best Practices and Tips

- AJAX Rules are supported on Search Flow and Model Configuration pages.
- AJAX rules are not supported for rules that have arrays as condition/action attributes or action inputs.
- AJAX rules are not supported for the default wizard.
- If an AJAX rule fails to evaluate properly, the action will be timed out in 2 minutes.
- The HTML structure has been modified and may not work well with any custom JavaScript on the page. Please validate upon upgrading to 11.1.

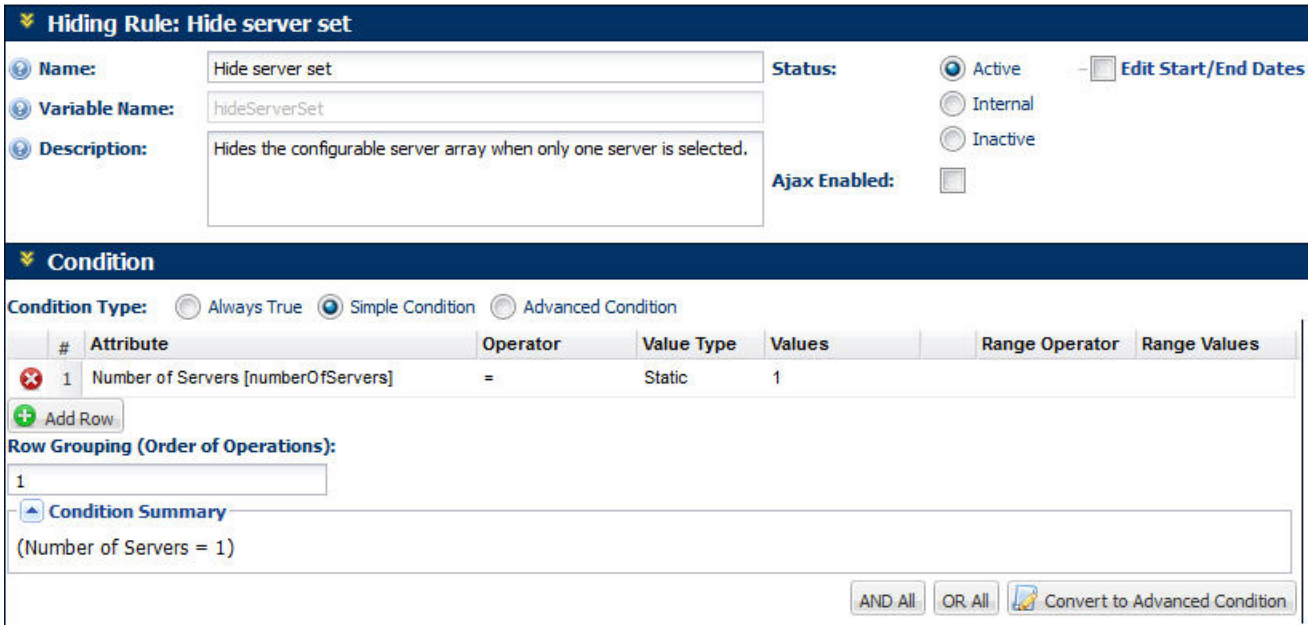
- AJAX Rule without SOAP/ Table Queries –AJAX rules are not recommended to be used on rules with a large amount of BML, SOAP calls, or table queries as it could adversely impact performance and hence, end user experience.
- AJAX Rule with fewer action attributes - AJAX rules update the page using javascript post page loads. Hence it is important that we limit the number of attributes being affected per AJAX.
- Group headers that are hidden on page load cannot be shown/hidden again via AJAX. While attribute in the group can be shown/hidden via an AJAX hiding rule, the group header will remain hidden until the user performs an Update action.

ADMINISTRATION

Creating a Hiding Rule

Follow these steps:

1. Navigate to your Product Family > Hiding Rules> List
2. Click **Add** to create a new rule.
3. Enter the **Properties** of the rule.
4. Select the **Condition Type**.



5. Choose to apply the rule to either **Configuration, Selection or both**. *Configuration is the default selection.*
6. Click **Add Attributes** to open the Action Attribute window.

7. Choose the **configurable attributes** that should be hidden when the condition is met and click **Add**.



Adding an Associated Recommendation Rule

A hiding rule can have one associated recommendation rule.

When users are configuring a model, they could end up with an internal constraint error, which they cannot fix. An internal constraint error occurs when there are constraints on hidden attributes and the values of these hidden attributes cannot be modified. To avoid these situations, you can modify the values of hidden attributes through recommendations. In this case, if a constraint is fired, then the value of this attribute can be set by the recommendation to avoid an internal constraint error.

You are unable to set the value into an action attribute because it could be a "None" set type and this will prevent the user from continuing on with the configuration. To avoid this, we provide attribute set types in recommendations so that the recommendation rule set type takes precedence over the attribute set type and modifies the value of the hidden attribute. Therefore the constraint action becomes invalid and hence the internal constraint error is no longer visible.

Creating an Associated Recommendation Rule

Follow these steps:

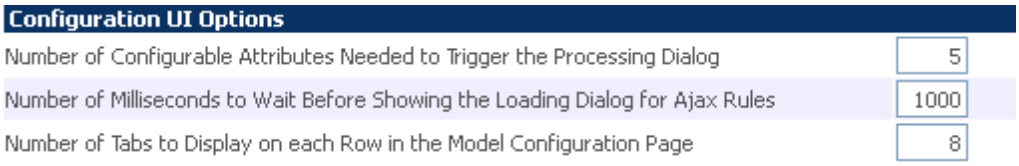
1. Click Add Associated Recommendation Rule from the right panel of the Action section of a Hiding Rule.
2. Enter the Properties of the rule.
3. Select your Condition Type.
4. Choose to apply the rule to either **Configuration, Selection or both**. *Configuration is the default selection.*
5. Choose the Action type:
   - **Standard:** You will use the UI to add attributes.
   - **Table Lookup**: Select attribute values from a table. See Table Based Rules for more information.
6. Choose an attribute whose value needs to be set. Under Values to Set:
   1. Choose attribute values from the **Select Values** menu, OR
   2. Click **Define BML Function** to set values through a BML script.\

7. Choose the **Set Type**:
    - **None:** The user can input any value without encountering an error message.
    - **Set:** A recommendation rule can change the value entered by the user unless the attribute is locked by the user.
    - **Forced Set:** A recommendation rule will change the value of the attributes regardless of what the user enters and if the attribute is locked or not.
8. Click **Update and Back** to save and return to the Hiding Attribute Rule where you will now see a link to the associated rule.

Enable AJAX

Follow these steps:

1. Navigate to the **properties** section of the rule.
2. Select the **AJAX Enabled** box.
3. Determine if you'd like to show or hide the processing dialog box for AJAX rules. *This is independence of the processing dialog box for the update action.*
    1. Navigate from Admin Home Page > Products > Configuration Settings.
    2. Enter a value for **Number of Milliseconds to Wait Before Showing the Loading Dialog for AJAX Rules**:
        1. **A positive number**: Sets how long the dialog wil appear (in milliseconds).
        2. **A negative number**: Disables the processing dialog for AJAX rules.
        3. **Zero (0):** Dialog box will appear instantaneously.



NOTES



By allowing the fullaccessuser user to disassociate a rule, we are providing them with the option of being able to easily delete the associated rule.

To "dynamically" hide attributes, make sure the condition attributes are set to auto-update.

All Hiding Rules (AJAX and non-AJAX) will collapse the space taken by a group when all attributes in

the group are hidden. This functionality will only be available for custom groups (non-default).

The Associated Recommendation Rule is available on the Recommendation Rules List page.

The associated recommendation rule will not be updated at the time of updating the parent rule, but a link to it is provided so that it can be updated after modifying the hiding rule.

If the Hiding Attribute rule is deleted, the Associated Recommendation Rule is NOT deleted. It can be accessed on the Recommendation Rules List page and continues to fire if the condition is met. If the Recommendation Rule is deleted, then the Hiding Attribute rule is not deleted. It can be accessed on the Hiding Attributes Rules page and continues to fire when the condition is met.

RELATED TOPICS

TABLE BASED RULES

OVERVIEW

BigMachines provides a point-and-click editor to define recommendation and constraint rules that access data tables, without resorting to BMQL. **Table based-rules allow for:**

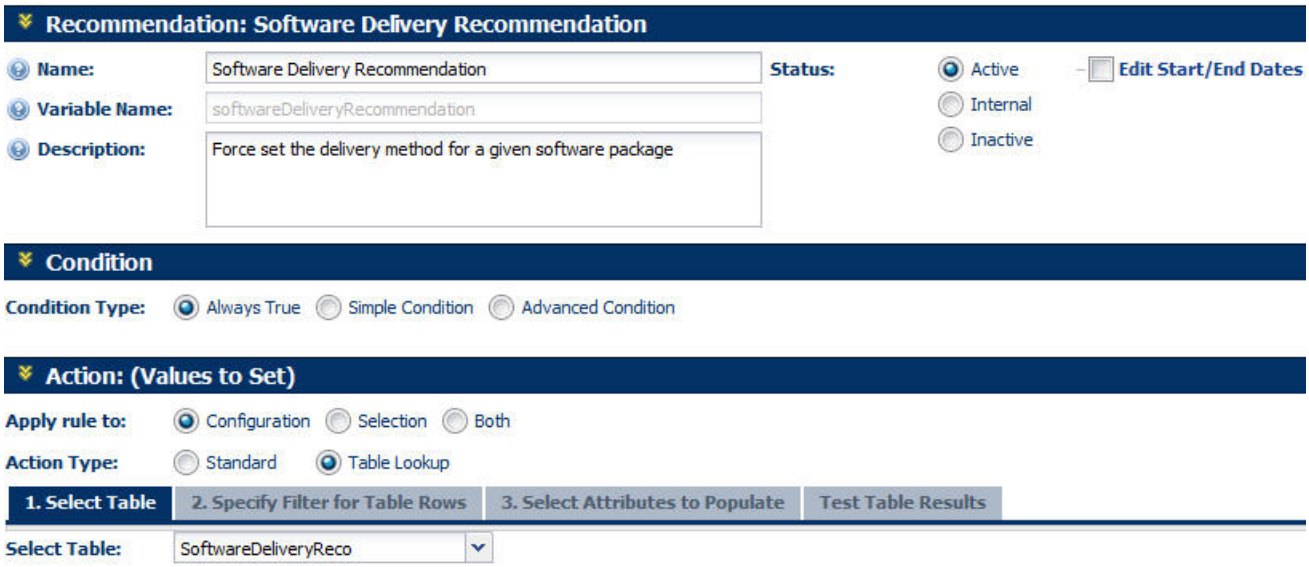
- Ability to return attribute values along with table data
- Query Debugger
- Point and Click UI for BMQL

Table based rules fire on the user-side when the condition is met. However, if no results are returned from the query then there is no impact on the action attributes. When multiple rows are returned for an action attribute, all returned values are recommended or constrained.

ADMINISTRATION

Creating a Table Based Rule

1. Using the Configuration Quick Links, navigate to Product Family > List of Recommendation Rules.
2. Click **Add** to create a new rule.
3. Enter the **Properties** of the rule.
4. Select the **Condition Type**.
5. Choose to apply the rule to either **Configuration, Selection or both**. *Configuration is the default selection.*
6. Select **Table-Based** as your Action Type. This will open a field of four tabs.



If you are creating a Recommended Item Rule, there will be an option Split Items.

7. Select the **Data Table** from which you would like to retrieve data, in the first tab,
8. Click the **Specify Filter for Table Rows** tab and select to Evaluate the Row with:
    - **Static Entry:** You define a static value under the Value header.
    - **Attribute:** You select a configurable attribute to map the Table Column to in a drop-down menu that appears under the Value header.

Array attributes used in the filter will execute for each array index. Also, array attributes with the same control attribute can be used in the same filter.





9. Enter the order of operations in the **Row Grouping** section.

If you only have one condition, you will still need to enter the value of 1 in this field.

10. Click into the **Value** cell and select attribute values from the drop-down menu or enter a static value. This is based on what you selected in Step 8.
11. Click on the **Select Attributes to Populate** tab.

12. Click **Add Row** to add:
    - The attribute(s) whose value you will populate with table data.
    - The table column that will populate the attribute
    - Set Type (None, Set or Forced Set), and
    - Message Type (Static or from a Table Column).



For Recommendation Rules: Specify the action attribute and the table column from which the recommendation value will be chosen. In addition, select a Set type for the rule.

For Constraint Rules: Specify the action attribute(s) and the constrained value. For menu-type action attributes, choose the constraint type: Constrain or Allow.

13. Click the **Test Table Results** tab.
14. Select the **Table Column and Value** you'd like to test and then **click Run**.



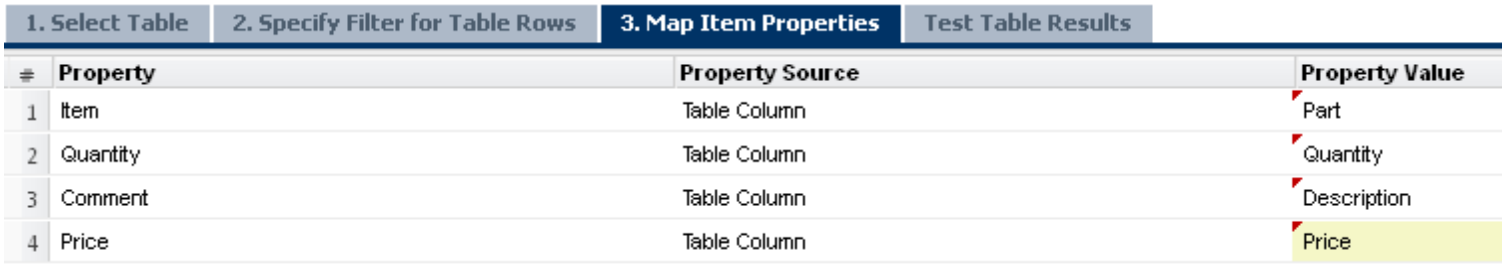
What is returned as an Ouput is the information from the table that will be returned when a user selects Meat Lovers as the Specialty Type (in this case).

Creating a Recommended Item Table Based Rule

Creating a Recommended Item Table Based rule is similar to how you'll use tables in Recommendations and Constraints. The main difference is instead of the 3rd tab being Select Attributes to populate, you'll have a Map Item Properties tab.

1. Follow **Steps 1-10** from Creating a Table Based Rule.

2. Click the **Map Item Properties tab**. Here you will map attributes, static values or table columns to the item, quantity, comment and price (i.e. return string columns).



#	Property	Property Source	Property Value
1	Item	Table Column	Part
2	Quantity	Table Column	Quantity
3	Comment	Table Column	Description
4	Price	Table Column	Price

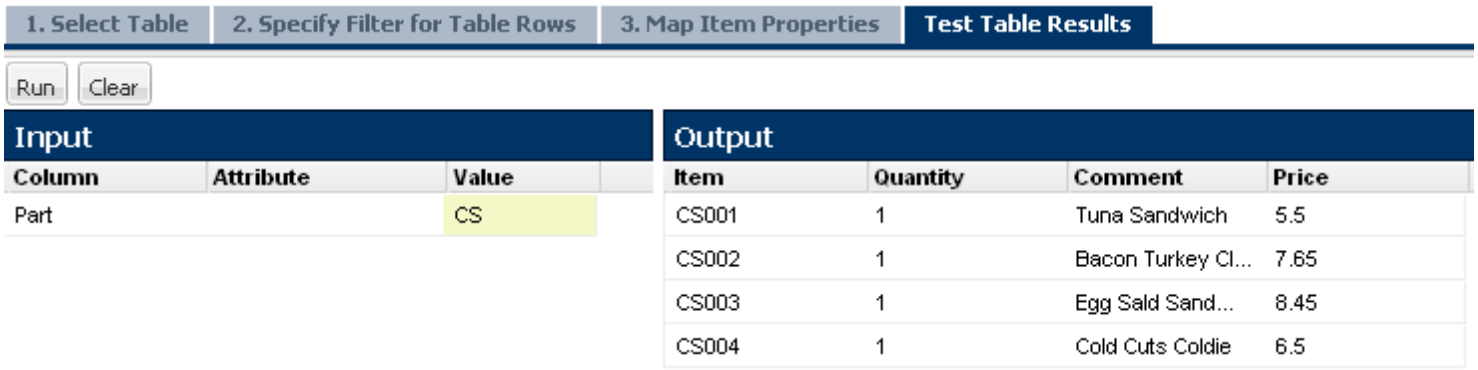
HTML and Date attributes cannot be used as inputs for Map Items Properties.

Array attributes can only be mapped to a return column when at least one array attribute with the same control was specified in the filter. The same index used in the filter will also be used to populate the mapping results

Property (Rec Item Display Columns)	Property Source Option	Property Value
Item	Table Column	You must map directly to a table column. For example, "Part".
Quantity	Table Column Static Entry Attribute	You can map to a table column, you can define a value or you can map it to a configurable attribute. **Note: If you map the quantity field to a Table Column,the table column MUST be of an integer data type.**
Comment	Table Column Static Entry Attribute	You can map to a table column, you can define a value or you can map it to a configurable attribute.
Price	Table Column Attribute Item Master	You can map to a table column, you can map to an attribute or you can map directly to the items master (parts database).

3. Click the **Test Table Results tab**. In the Input Section, the Table Column will already be set (and read-only).

4. Enter the **Value** you'd like to test and then click **Run**.



Action Attributes in Table Based Rules

The supported action attributes are Text, Integer, Float, Currency, HTML and Data type configurable attributes.

Attribute Type	Data Table Column Type	Supported Values	Comments
Text	String	All string values	
Integer	Integer	All integer values	
Float	Float	All float values	
Date	String	Values in database date format: yyyy-MM-dd hh:mm:ss	
HTML	String	All string values	Not supported in constraint rules.
Currency	Float	All float values	For multi-currency applications, exchange rate is applied on the values returned for the currency attribute.
Text Single Select Menu	String	All text values	
Integer Single Select Menu	Integer	All integer values	
Float Single Select Menu	Float	All float values	
Text Multi Select Menu	String	All text values	Multiple values can be recommended or constrained if '~' delimited values are returned.
Integer Multi Select Menu	Integer	All integer values	
Float Multi Select Menu	Float	All float values	

Array Attributes in Table Based Rule Actions

Recommendation and Constraint table-based rule functionality allow arrays to be used in the actions. You can use recommendation rules to set values or use constraint rules to constraint values of individual array indexes. You can use the value(s) of an array set to

pull information from a data table by mapping an array attribute to a data table column. The column will be used in the filter to evaluate the array attribute value. For a single array attribute, each index will be evaluated individually against the data table.

Notes about Recommendation Rules:

- If there are no array attributes in the filter tab, the results will be applied to array attributes at all indexes in the result tab.
- Set types will remain the same as current recommendation rules (None, Set, Force Set).
- No match for a particular index will leave the index value and the set type 'as is'.
- The first row that matches the filter will set the array attribute index.

Notes about Constraint Rules

- Constraint behavior for each array attribute will be the same as current constraint rules (Allow, Do Not Allow).
- If there are not matches from the data table, for a particular index, constraints will not be applied to that index.
- If no array attributes are in the filter that calls the data table, then all indexes of the array attribute will be constrained based off of the returned value.
- The filter will be compared against all rows in the table and all matches will constrain the related index.
- If you specify only non-array attributes in the Table Query Condition, the action will apply to all indexes of the specified array attribute.

Notes about Behavior

Simple table based rules operate per index; they will not operate across indexes. Attribute type limitations are consistent with existing functionality. The table below represents how rules will operate depending on the condition and filters.

	Non-Array Attributes Only (in mapping)	Array Only Attributes (in mapping)	Array & Non-Array Attributes (in mapping)
Condition: Always True **Filter**: Static Values	Map first returned row to non-array attribute.	Map first row to each index of array. Same value for every index.	Map first row to each index of array and to the Non-Array.
Condition: Always True **Filter:** Non-Array Attributes	Map first returned row to non-array attribute.	Map first row to each index of array. Same value for every index.	Map first row to each index of array and to the Non-Array.
Condition: Always True	All indexes run filter; The recommended	All indexes will run filter; Each index applies the first row of the index's filter results; Only	All indexes will run filter; Each index applies the first row of the index's filter results; Only indexes with results will perform the

	Non-Array Attributes Only (in mapping)	Array Only Attributes (in mapping)	Array & Non-Array Attributes (in mapping)
Filter: Array Attributes	value will be the last array index value.	indexes with results will perform the recommendation or constraint.	recommendation or constraint. Non-Array attribute is set to the last result returned by the last index.
Condition: Non –Array Attributes **Filter:** Static Values	Map first row returned to Non-array attribute.	Map first row to each index of array. Same value for every index	Map first row to each index of array and to the Non-Array.
Condition: Non-Array Attributes **Filter:** Non-Array Attributes	Map first row returned to Non-array attribute.	Map first row to each index of array. Same value for every index	Map first row to each index of array and to the Non-Array.
Condition: Non-Array Attributes **Filter:** Array Attributes	All indexes run filter; The recommended value will be the last array index value.	All indexes will run filter; Each index applies the first row of the index's filter results; Only indexes with results will perform the recommendation or constraint.	All indexes will run filter; Each index applies the first row of the index's filter results; Only indexes with results will perform the recommendation or constraint. Non-Array attribute is set to the last result returned by the last index.

NOTES



Only "=" and "Not=" operators are supported for querying blank values on text type attributes. If any other operator is used for blank values, the query will assume that the operator is "=" for "Contains", "Start With" and "ends With". Similarly, the query will assume the opertator is "Not=" for "Not Contains", "Not Starts" and "Not Ends with" when querying for blank values.

The user can specify multiple action attributes in the Select Attributes to Populate tab for any table-based rule. This behaves similar to standard configuration rules, which can have multiple action attributes where the rules fires per action attribute and not all action attributes have to be present in the flow for the rule to fire for one particular attribute. The rule fires if the condition is met and the condition attribute and the action attribute are in the flow.

All table data must be deployed to see the correct results in the Debugger.



Duplicate values returned by the query are removed and only unique values are recommended or constrained.

When deploying configuration, you can deploy data tables that are used in Table Based Rules, which have undeployed changes.

You can include data tables that are used in table-based rule in configuration test to production. If the data table does not exist in production, it will be uploaded to the Default folder. If the data table exists in production, then the test site data will be appended to the production site data for that table.

If multiple messages are returned, then all messages are displayed in a comma delimited list to the end user, no matter the value chosen by the user.

Array attribute type limitations are consistent with existing functionality. For example, if the table column is of numeric data type, then the array attribute(s) must be numeric.

In cases where quantity <=0, quantity will be shown as Qty = 1.

Prices will show as the following in these situations:
o On Request<=0 : Non Number value that is not the keyword “Base”
o Not DefinedNo price at all, or blank
o Default Unit PriceKeyword “Base” is found in the table



Blank queries are not supported for numeric data types.

Simple table based rules operate per index; they will not operate across indexes.

Only array attributes from the same set can be used in the Filter and Mapping tabs. Mix and match of array attributes across sets is not allowed.

RELATED TOPICS

PRICING RULES

OVERVIEW

What are they?

Pricing rules are used to calculate price based on how a product is configured. Pricing rules can be based on a combination of one or more configured values. Like most rules, a pricing rule also has a condition and an action. The action determines the price of the model being configured.

How do they work?

Displaying the price to the user is the default setting, though doing so is optional. If the price should be shown to the user can be determined by an Advanced Function. If the function returns true then the price is displayed. If the function returns false, then the price is not displayed. Including the price in the total configured price is also optional. The default value is true, which indicates to the system that the price should be included in the total configured price. An advanced function can modify a default value if necessary.

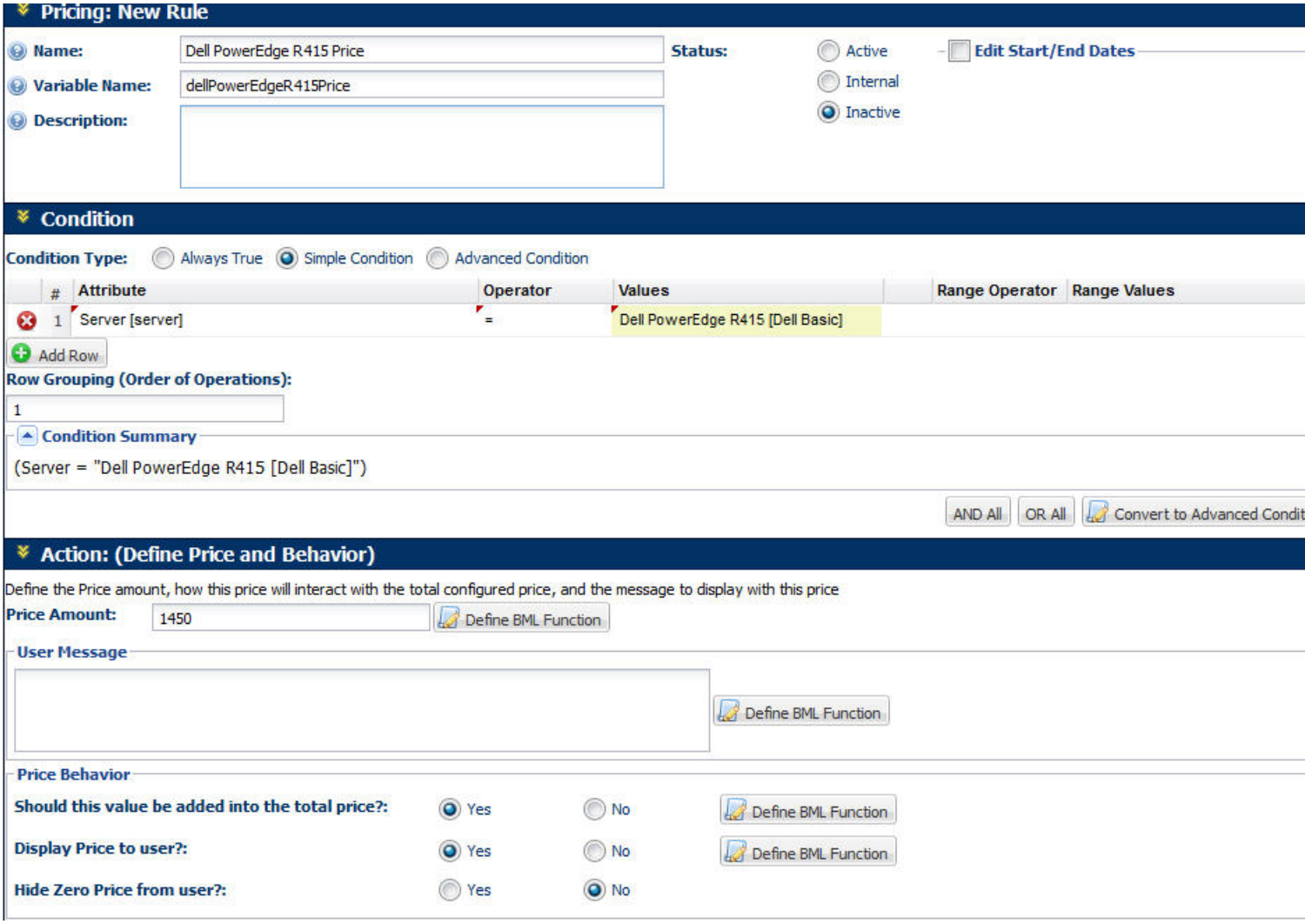
ADMINISTRATION

Managing Simple Pricing Rules

Example: You need to set the price for a Dell PowerEdge R415 Server. The price is $1,450.00.

1. Navigate from: Product Family > Prices> List
2. Click **Add** to create a new rule.
3. Enter the **Properties** of the rule.
4. Select the **Condition Type**.
5. Define the **Price Amount** by entering a static value.
6. Enter a **User Message,** which will be displayed with the price.
7. Define the **Price Behavior** by choosing **Yes or No** for the following options:
    - Should this value be added to the total price?
    - Display Price to user?
    - Hide Zero Price from user?[1]

[1]If no, then a value of $0 will be displayed. When the user decides to place an order and they select Soup of the Day, the price will be included with the total configured price of the model.



USE CASES

Advanced Pricing Rule

Scenario: You will be setting the prices for the pizzas based on their size and crust type. *You will also need to create a user message that explains to the user exactly what they are ordering*. According to the menu, you have the following options and prices:

Size	Crust Type	Price
Small	Thin Crust	$9.00
Medium	Thin Crust	$12.00
Large	Thin Crust	$15.00
Medium	Deep Dish	$17.00
Large	Deep Dish	$20.00

1. Navigate from: Product Family > Prices> List
2. Click **Add** to create a new rule.
3. Enter the **Properties** of the rule.
4. Select the **Condition Type**



This rule can run all of the time, so you should select Always True.

5. Define the **Price Amount** . Since you are creating a rule for multiple items, you will need to create an Advanced Function by clicking Define BML Function.
6. Choose the Attributes you will use the script by clicking the Add Attributes button in the function editor. *In this case, we'll select Size and Crust Type.*
7. Begin to write the pricing script in BML.
8. You will need to first assign a variable to zero. Try **price = 0**;

```
price=0;

//Setting price for small pizza
if(size=="Small"){
        price=10;

                //Setting price of Medium Thin Crust Pizza
                }elif(size=="Medium" AND crustType=="Thin Crust"){
                        price=12;

                //Setting price of Medium Deep Dish Pizza
                }elif(size=="Medium" AND crustType=="Deep Dish"){
                        price=17;

                //Setting price of Large Thin Crust Pizza
                }elif(size=="Large" AND crustType=="Thin Crust"){
                        price=15;

                //Setting Price of Large Deep Dish Pizza
                }elif(size=="Large" AND crustType=="Deep Dish"){
                        price=20;
}

return price;
```

9. Set up the User Message that the buyer will see when they see the Price. Click the Define BML button.

10. Select the necessary attributes. They should be the same as found in Step 6.
11. Create an empty string variable to hold your user message string.
12. Write the user message to be returned.

```
message = "";

message = "Price for a ";
message = message + size + " " + crustType + " " + "Pizza";

return message;
```

Negative Pricing

Before implementing Negative Pricing, review your requirements to ensure that this is absolutely necessary. *Ideally, this situation should be avoided.* Business functional requirements that make Negative Pricing appealing:

- **Trade-ins:** A previously ordered item is swapped for another.
- **Upgrades:** A previously ordered item is exchanged for a newer model.
- **Renewals:** A previously ordered item is upgraded when renewed.

Recommended Solutions	Actions
Set values of items w/ a negative price to $0	1. Create an additional Pricing Rule that shows the negative price, but doesn't pass the price to Commerce. 2. Store the negative price in a hidden or Recommended Item "Comments" field. 3. Update the price to be a negative price in Commerce. *A new Commerce line-level float field must be created to support this price, because the* `standard_pricelist_priceeach` *doesn't support negative prices.*
Hide Recommended Item prices in Config	Use a pricing rule to show the Configuration grand total price, not just the negative price.
Store the negative price in a data structure or Attribute	**Do one of the following:** • Use the Recommended Items "Comments" field. • Use an Array. • Use a text field with delimited data.
Alternate display	If an Array is used to store item prices, it could be made visible in the Configuration Flow, so users can see the price per item.



Recommended item rules do not support negative prices. *Returning a negative price for an item will return a price of $0.*

NOTES



During reconfiguration, pricing rules are always evaluated, regardless of whether the condition attributes are in the flows.

In the getPrice() SOAP call, the Hide Zero Price behavior is replicated by using the attribute bm:hideZeroPrice.

RELATED TOPIC

PRICE BOOKS AND CONFIGURATION RULES

OVERVIEW

Price books can be used in configuration rules as a variable in rule conditions and actions. For example, you can use a recommendation rule to force set a price book or constrain a price book. Constrained price books are removed from the price book list unless they are currently selected, in which case they will appear marked. You could also use a price book as a rule condition for a hiding rule and hide particular attribute(s) when a certain price book is selected. Please see the article Price Books and Configuration Rules for more information.

There are many ways in which the end user can find themselves interacting with price books that are rule inputs of configuration rules. Here are a few of the common rules they might come across:

Recommendation Rule

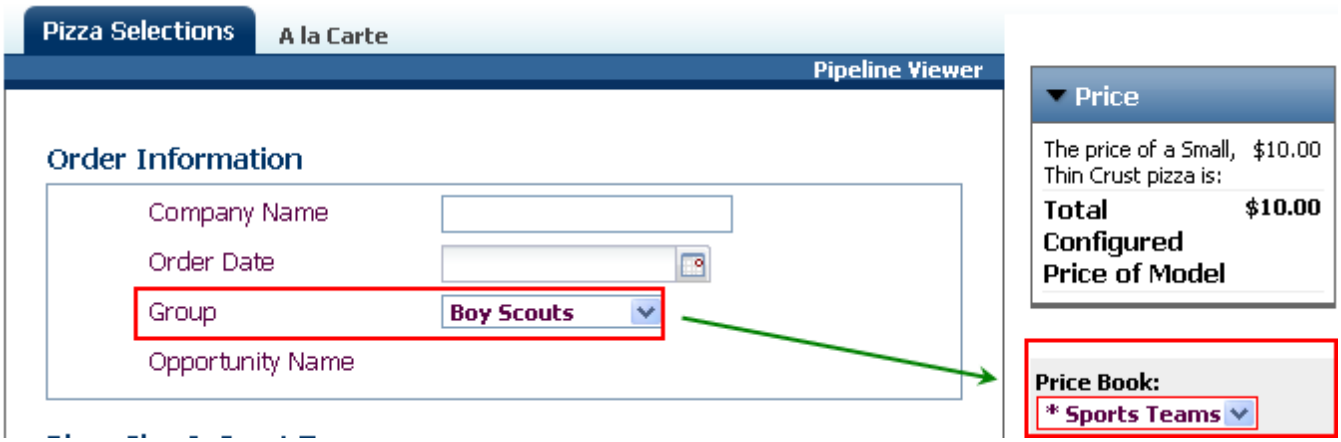
When a price book is used as the action attribute in a recommendation rule, it will be forced set for the user, meaning they will not be able to change it. In the example below, a recommendation rule is running that says when Girl Scouts is selected as the Group, then the Price Book should be set to Scouting. Notice that it is read only.



Constraint Rule

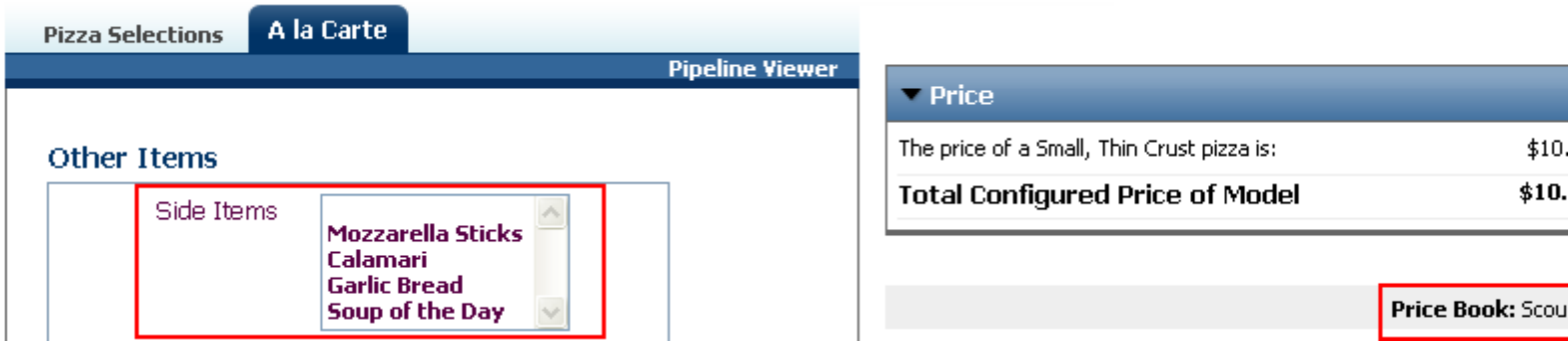
When a price book is used as the action attribute in a constraint rule, it will be removed from the price book list unless it is currently selected. In the example below, a constraint rule is running that says when Boy Scouts is selected, the Price Book should be set to Scouting. Notice that the current price book is Sports Teams and is marked with an asterisk (*) and is encased in red.

When the user clicks on the Price Book list to select a new price book, you'll notice that the only one available for selection is Scouting.

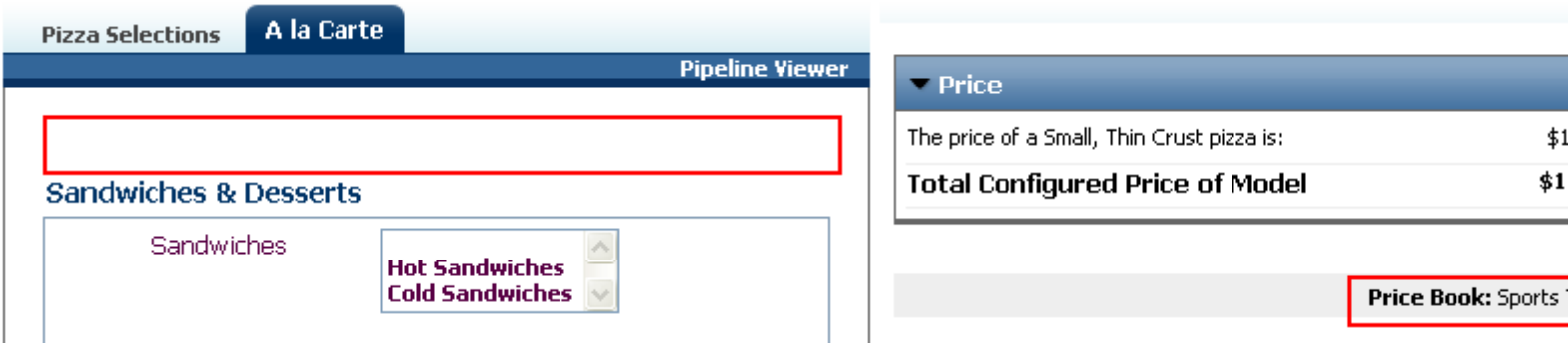


Hiding Rule

When a price book is used as a rule condition in a hiding rule, it can hide attribute(s) based on which price book is selected. In the example below, you'll notice that in the first image, you can see the Side Items attribute displaying.



However, when the price book is changed to Sports Teams, you'll notice that the attribute has been hidden from view.

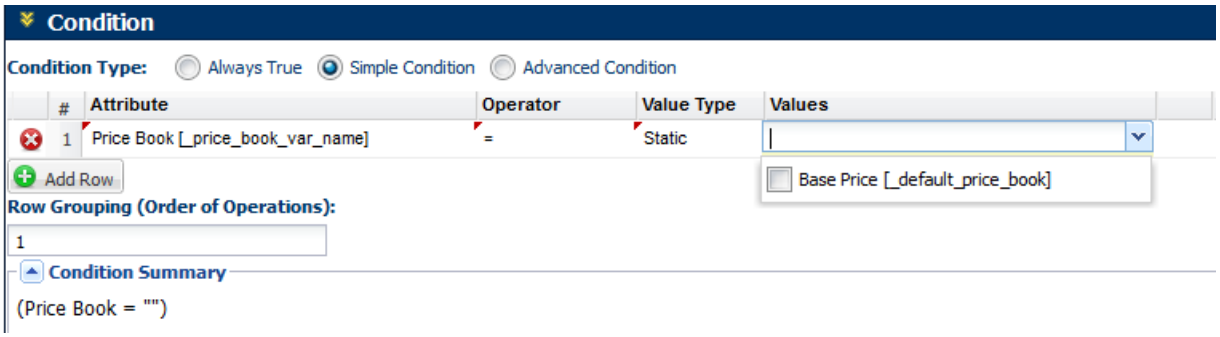


ADMINISTRATION

As previously mentioned, price books are now available for use as rule inputs, whether as a rule condition or as an action attribute. They can also be used in BML scripts. The following steps will explain how to use a select a price book as a rule input.

Using Price Books as a Rule Condition

1. Navigate to a **configuration rule**.
2. To use a price book as a rule condition, select **Simple Condition**.
3. Select **System Attributes** from the folder drop-down menu.
4. Select **Price Book** from the options.
5. Select the appropriate value(s) for the price book.



NOTES

Price book information is not set when the items is added to Favorites, even if a specific price book has been recommended. On reconfigure, the recommended price book will be set and the user will be allowed to change it.

Upon Add from Catalog/Reconfigure in Commerce, the transaction price book will take precedences over price books returned from a recommendation rule.

The price book variable name is _price_book_var_name.

RELATED TOPICS

BILL OF MATERIAL (BOM) RULES

OVERVIEW

What are they?

A configurable BOM (Bill of Materials) is a set of parts specific to a certain product configuration. For each configuration, where a product has a different set of parts, you can create a BOM.

How do they work?

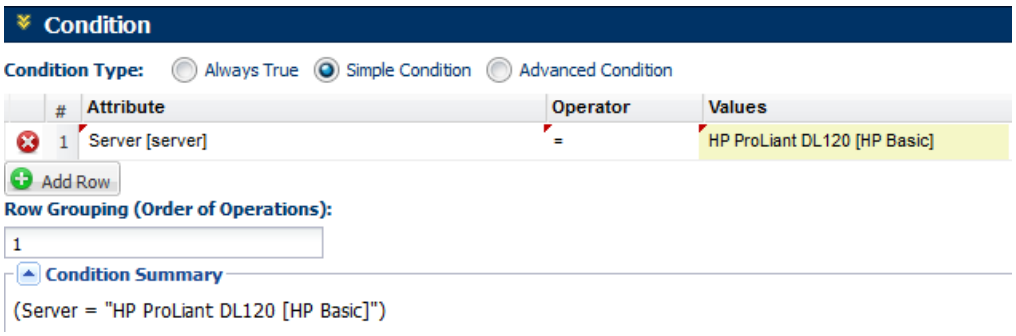
When a BOM triggers, it appears to buyers on commerce document line item pages. If the parts contained in the BOM link up with the parts in the parts database, then parts pricing also appears on line item pages. Buyers see multiple BOMs when several apply to a given configuration. *When the system triggers multiple advanced BOMs, which are created as strings, all of the strings appear concatenated in the order they were generated.*

A BOM rule has a condition and an action. The values of the attribute(s) selected as the condition attribute(s) determine the result of the condition, which when true triggers an action. The action here is to display a list showing the parts description and their quantities. This list is shown in a separate tab in the pipeline viewer or in the commerce document.

ADMINISTRATION

Adding a new Rule

1. Navigate to your Product Family > Bill of Materials > List
2. Click **Add** to create a new rule.
3. Enter the **Properties** of the rule.
4. Select the **rule conditions**:



5. Set the rule **action**:
    1. **Simple Bill of Materials:** Select if you know the parts you need to add. Click Add Row to add parts to the BOM string.
    2. **Advanced Bill of Materials:** Select to write a BML script that assembles a BOM string. Click Define BML Function to create the script.



The advanced function returns a delimited string. For example: MS001~1~IS001~3 (Part1~Qty1~Part2~Qty2)

6. Verify the rule is running by checking the Pipeline Viewer in Configuration.

BOM Best Practices

1. Define Configuration options and values (e.g. drop-down menus).
2. Define BOM modules. *This can be challenging.*
3. Link BOM module part number(s) to option values. *This typically takes the most time.*
4. Write BigMachines BOM rules to query tables and return part numbers. *Can load tables to BigMachines from Excel.*

RELATED TOPICS

SELECTING CONDITION ATTRIBUTES

OVERVIEW

A rule is composed of a condition and an action. The condition is evaluated first. If the condition is evaluated to true, then the rule is fired. That is, an action is performed.

Rule Condition types

- **Always True**: The rule always fires.
- **Simple Condition:** Simple Conditions can be easily set up without the need to resort to coding. The users simply have to select the necessary condition attributes and specify the values of the attributes for which the rule is fired. Several comparison operators are provided for this purpose.
- **Advanced Function:** The advanced function is meant for complex condition logic. The users build their logic using BML in the fuction editor.

ADMINISTRATION

Managing Simple Conditions

You can select attributes to use in simple conditions and advanced functions. You will use simple conditions when you are writing a rule that needs to run under certain conditions. Instead of selecting Always True, you will need to select the Simple Condition radio button in the Condition section of the rule.

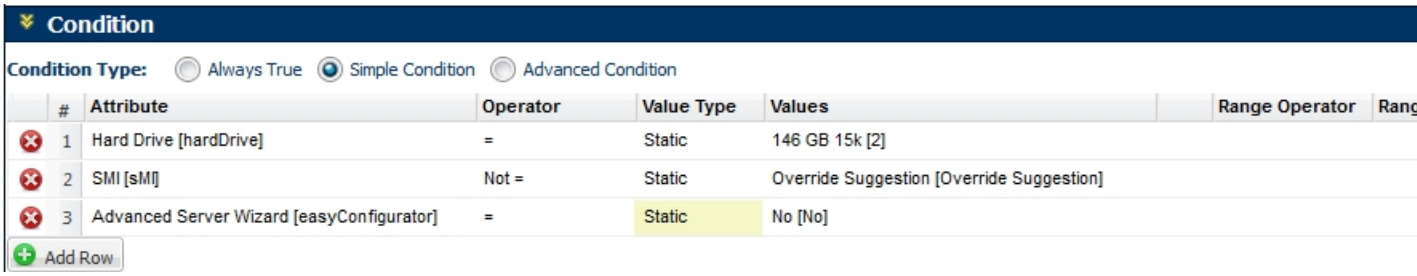
When using simple conditions, you have to select the necessary condition attributes and specify the values of the attributes for which the rule is fired. Several comparison operators are available for this purpose.

1. Click **Add Row** button.
2. Use the drop-down menu under Attributes to select your first condition attributes.
3. Choose an Operator from the drop-down menu.

Attribute Data Type	Available Operators
Text	• = • Not = • Starts With • Not Starts With • Ends With • Not Ends With • Contains • Not Contains
Currency, Float or Integer	• > • < • >=

Attribute Data Type	Available Operators
	• <= • = • Not =
Single Select Menu	• = • Not =
multi-select Menu	• = • Not = • Contains • Not Contains
Date	• = • Not = • Before • On or Before • After • On or After

4. Select the Value Type:
   - **Attribute:** Compare one attribute to another in a single condition row. Note: Attribute data type must be equal for the comparison.
   - **Static:** Enter a static value in the Values column.



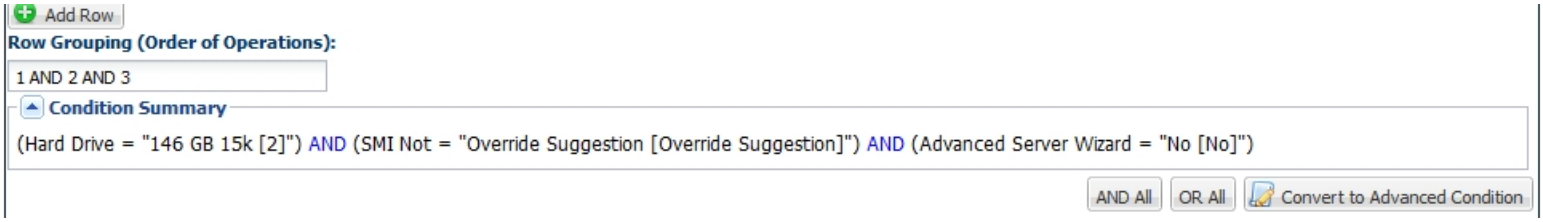
5. Choose the **value(s)** for the condition.



There may be a checkbox that is blank. You'll want to de-select this unless you want the rule to run for "none".

Simple Logic with Grouping

You can choose the order in which the statements are evaluated by using the Row Grouping section. The default evaluation happens from left to right. You can explicitly specify the order of operations with the help of parentheses. For example, the condition "1 AND 2 OR 3' is evaluated by default as (1 AND 2) OR 3. You can change the order of operations by typing in the row grouping section, 1 AND (2 OR 3). You can view the order of operations in the Row Grouping section (by entry row) or in the Condition Summary.



Row Grouping

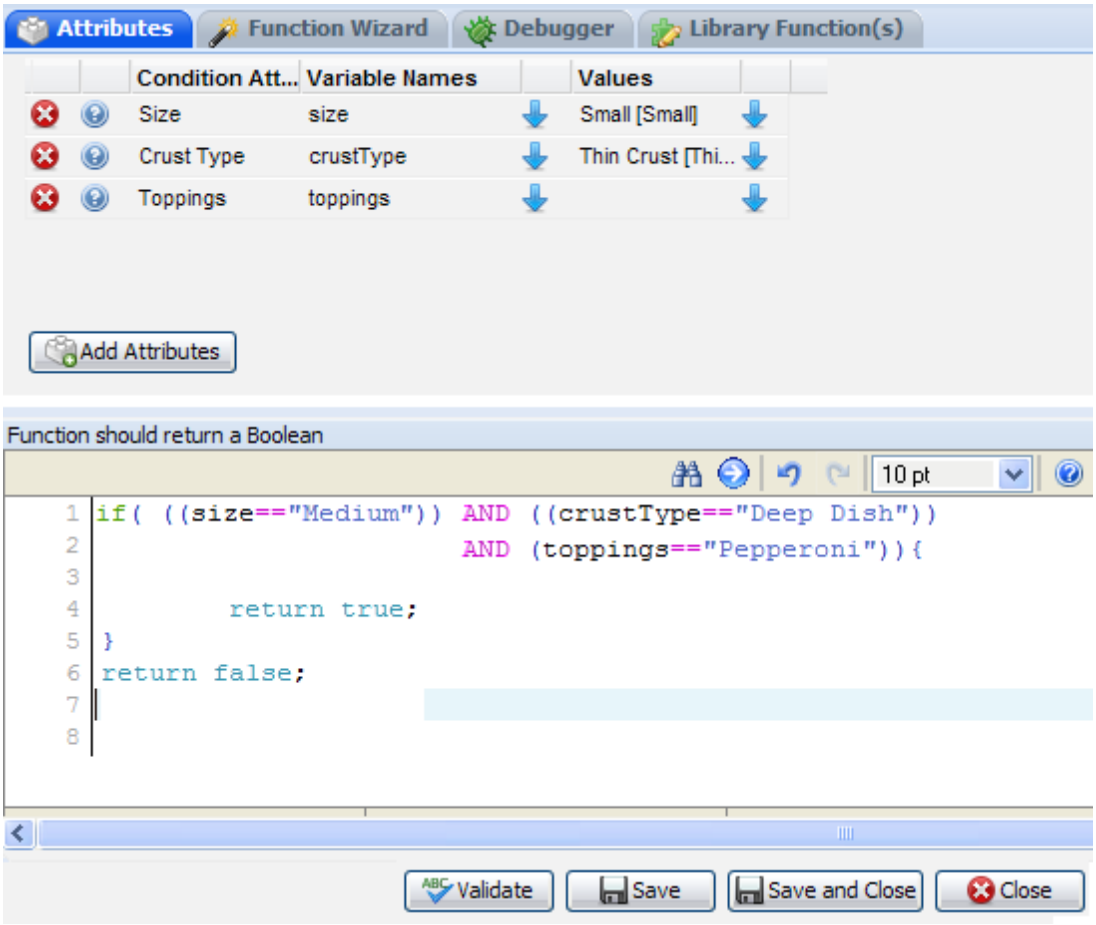
Row Grouping	Results	How to Achieve it
1 AND 2 AND 3	Rule fires when: myColor = "Red" AND myShape = "Circle" AND mySize = "Small"	Click AND ALL to achieve this result. **This is default logic.**
1 OR 2 OR 3	Rule fires when myColor = "Red" OR myShape = "Circle" or mySize = "Small"	Click OR ALL to achieve this logic.
(1 AND 2) OR 3	Rule fires when: myColor = "Red" AND myShape = "Circle" OR mySize = "Small"	Default parentheses from left to right. Assumes that first 1 AND 2 is evaluated and the result is OR'd with 3.
1 AND (2 OR 3)	Rule fires when: "myColor = "Red" AND myShape = "Circle" OR mySize = "Small"	The explicit parentheses in this statement changes the order the attributes are evaluated. At first, 2 or 3 is evaluated. The result of this is then AND-ed with 1.

Managing Advanced Conditions

The advanced function is meant for complex condition logic. You build the logic using BML in the function editor. You can access this by clicking the View/Edit the BML Function button.

Please refer to the section Developer Tools/BML for more information.

Sample of Advanced Conditions:



NOTES

When defining an array in the rule condition, you should be aware that the return types of the advanced action will match the return type of the configurable array being acted upon (string, int, float, etc).

For legacy rules, ranges are now removed and the condition enhanced to meet the use case.For example, instead of itemInMenu>=1 AND itemInMenu=>3, this will now be coverted to: itemInMenu = 1 OR 2 OR 3.

RELATED TOPICS

VIEWING PIPELINE VIEWER

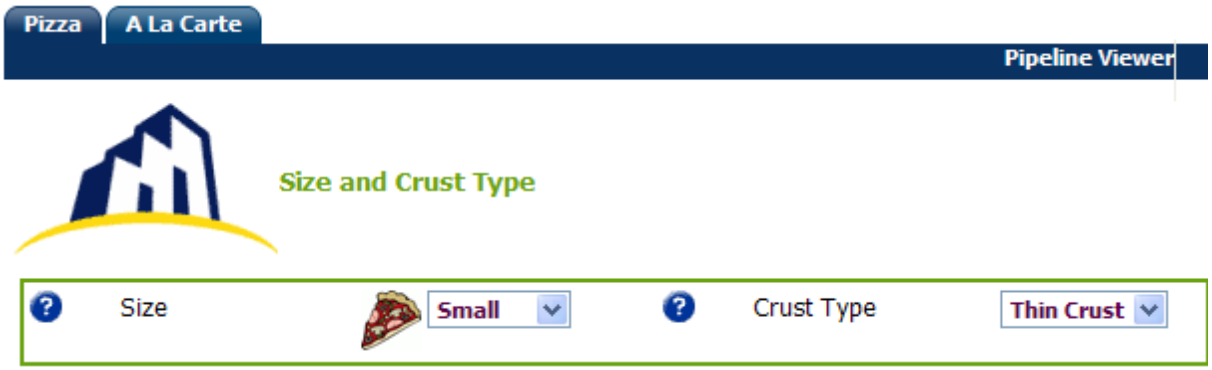
OVERVIEW

The Pipeline Viewer enables you to check a model's current attribute values and the rules applicable to the model. For example, it can tell you the following:

- If a rule has fired through AJAX (Rules tab)
- The currently active rules that each attribute present in the flow are inputs to (Attributes tab)
- All attributes that are inputs to all currently active rules (Rules tab)

Within Configuration

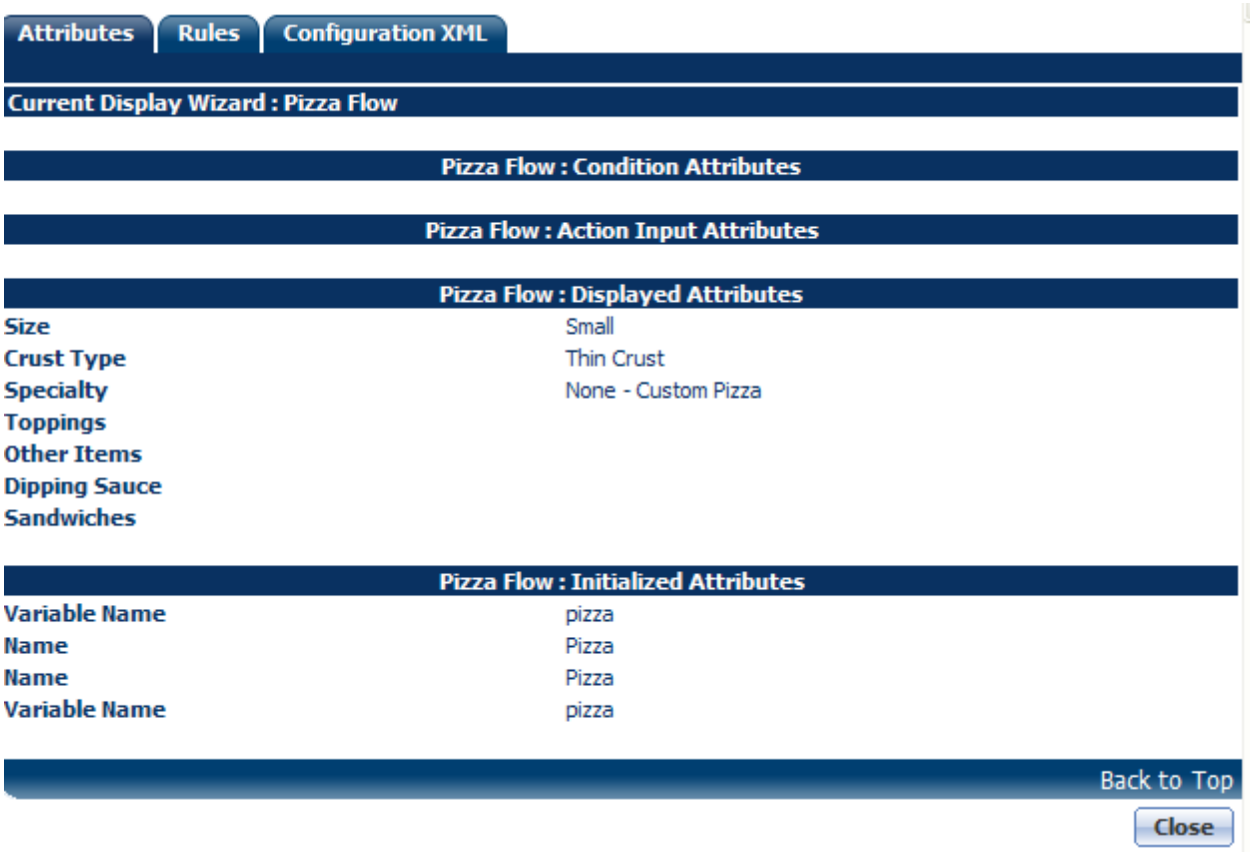
It is a tool for debugging and it is only available to fullaccessusers as a pop-up. The link to the pipeline viewer pop-up is found on the model configuration page.



Pipeline Viewer lists for each specific model

- **Attributes** : All the condition attributes, display attributes and their values based on the current model configuration page
- **Rules**: All the rules that are applicable for the model along with the rule types and the level of creation of each rule

- **BOM:** The Bill of Material associated with the model, the name of the BOM rule and the level of creation of the rule



When configuration cache is enabled, then all changes to configuration data must be deployed in order for it to appear in the Pipeline Viewer.

Commerce

A Pipeline Viewer is also available as a button within Commerce. This is only available to the superuser and fullaccessusers.



ADMINISTRATION

Pipeline Viewer Rule Precedence

Rule precedence according to the Pipeline Viewer

1. Recommendations
2. Constraints
3. Hiding Rules
4. Recommended Items

Actual Rule precedence

1. Config Flow Rules
2. Constraint Rules
3. Recommendation Rules
4. Constraint Rules (again)
5. Hiding Rules
6. Recommended Item Rules
7. BOM Rules

NOTES



The Pipeline Viewer pop-up does not dynamically refresh itself. Every time you change an attribute value and want to recheck how its value affects other attributes or rules in the flow, click the Pipeline Viewer link again.

If new configuration rules or flows do not appear in the Pipeline Viewer,ensure that either the configuration cache is disabled or that the segment was deployed so that the Pipeline Viewer is synchronized with current configuration data.

If a product does not have an associated BOM, the BOM tab in the Pipeline Viewer will not be available.



Edits cannot be made from within the Pipeline Viewer. All edits have to be made in the Catalog Definition section.

RELATED TOPICS

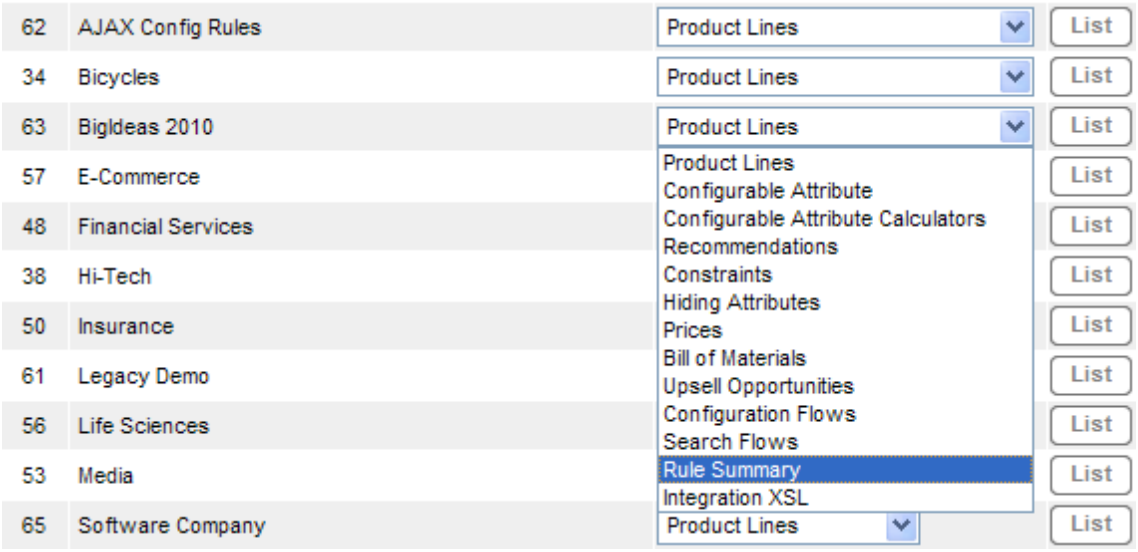
VIEWING RULE SUMMARY

OVERVIEW

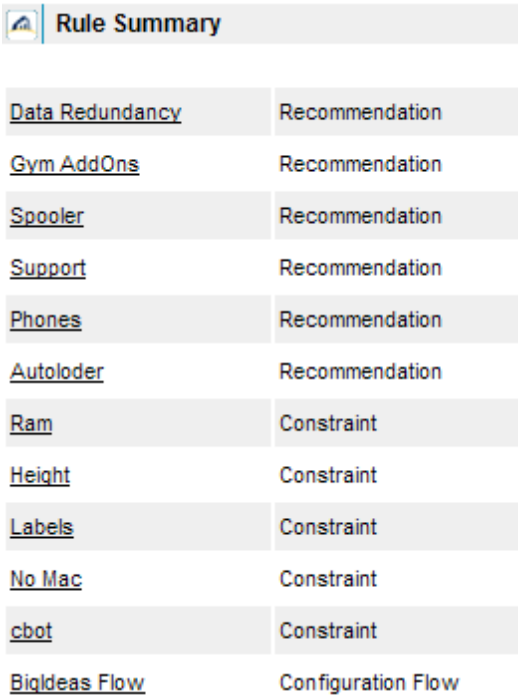
The rule summary is a report that shows the configuration rule created at a specific level of the product hierarchy.

ADMINISTRATION

1. Navigate from Admin Home Page > Configuration > Catalog Definition > Rule Summary > List
2. Navigate to the desired product hierarchy level (Product Family, Product Line or Model).
3. Choose **Rule Summary** from the drop-down navigation menu.



4. Click **List**. *The Rule Summary page lists the rules and their types.*



5. Click on a rule name to access a Rule Editor page.

RELATED TOPICS

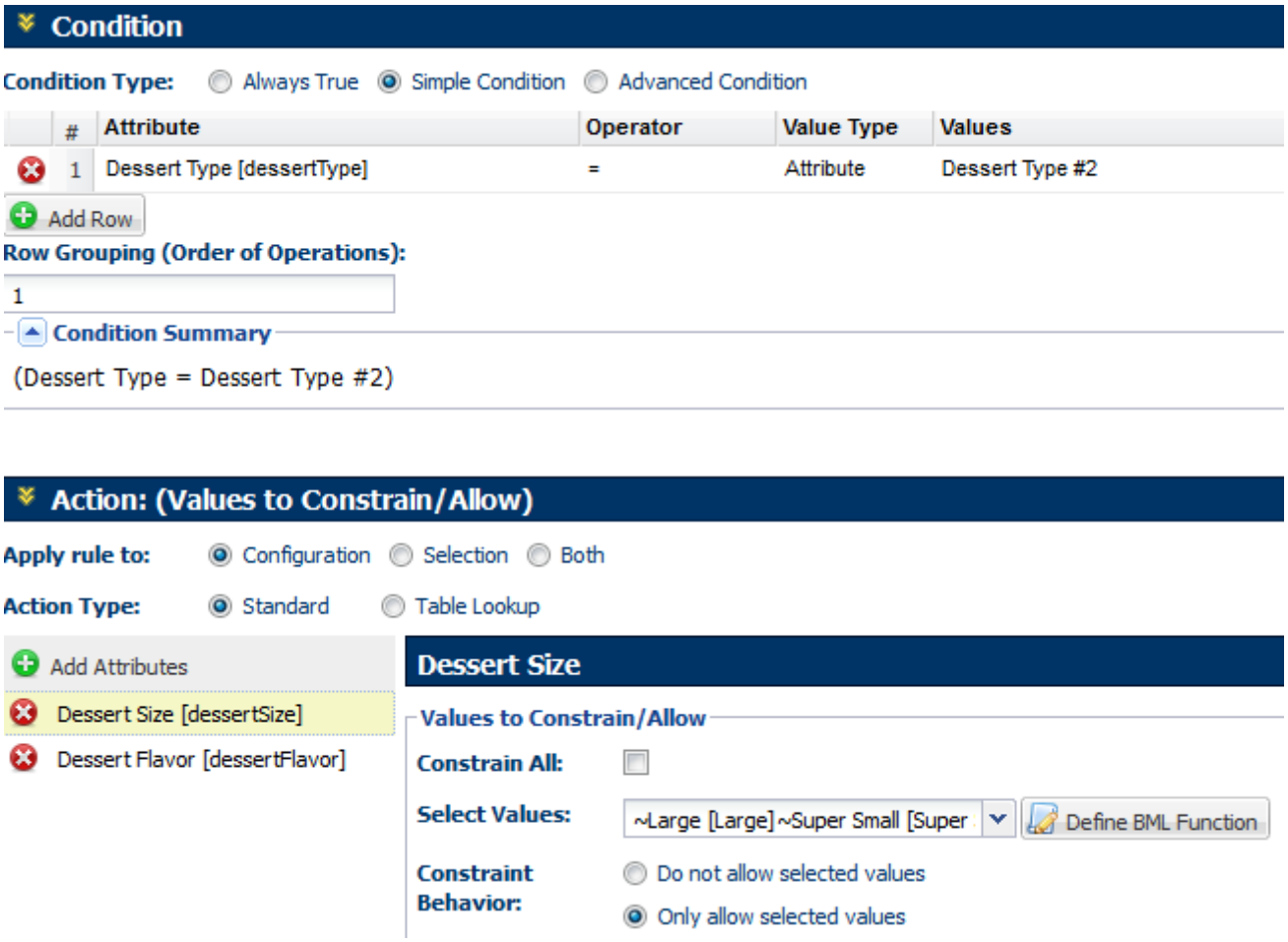
USING ARRAYS IN SIMPLE CONFIGURATION RULES

OVERVIEW

When rules are using arrays, they will operate on each index, as if each index is a separate rule. The rules are essentially breaking up array sets and treating them like individual attributes. When using rules to evaluate each row in the array, keep in mind:

- **For Hiding Rules:** The rules will hide array attributes on select indexes while leaving others visible.
- **For Recommendation Rules:** The rule will Force Set array attributes on select indexes and leave others unchanged.
- **For Constraint Rules:** The rule will constraint array attributes in select indexes and leaves others available.

Example



ADMINISTRATION

Rule behavior in various scenarios

	Simple Action on Array Attributes	Advanced Action on Array Attributes	Simple Action on Non-Array Attributes	Advanced Attributes on Non-Array Attributes
Simple Condition Contains Array	Condition will execute per index. Action will execute per index on indexes where condition is true.	Condition will execute per index. Action will execute per index on indexes where condition is true. Each Action will operate independently of the others and will execute when true; Return type of Advanced Action will match the selected action array attribute. Rule action will act on each index where condition evaluated to true.	Condition runs once, no change on operation.	Condition runs once, no change on operation.
Simple Condition Does Not Contain Array	Not Available	Operate same as previous versions. Return type will be an array type.	Operates the same as previous versions.	Operates the same as previous versions.
Advanced Condition	Not Available	Operate same as previous versions. Return type will be an array type.	Operates the same as previous versions.	Operates the same as previous versions.

USE CASES

Using Array Attributes in a Constraint Rule Condition that only allows certain values in the Action

In this use case, we'll be demonstrating how to use array attributes in the condition of a constraint rule that will only allow certain array attribute values in the action. This article assume you know how to navigate to constraint, hiding, and /or recommendation rules.

1. Enter the rule **Properties**.
2. Select a **Condition Type**. For this example, we'll select Simple Condition.
3. Use the drop-down menu under Attributes to choose your first condition (array-type) attributes.

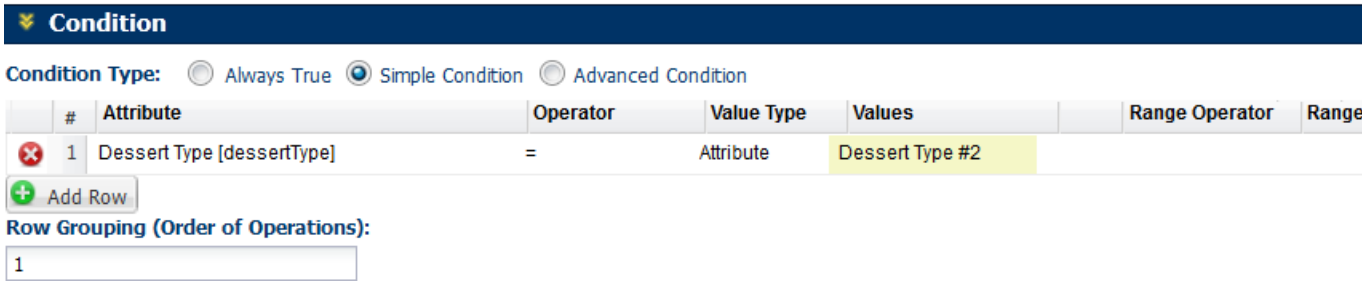


An array attribute must be used in the condition in order to use an array attribute, in a simple way, in the action. Also note, only array attributes that share a control attribute can be used when a simple condition contains an array attribute.

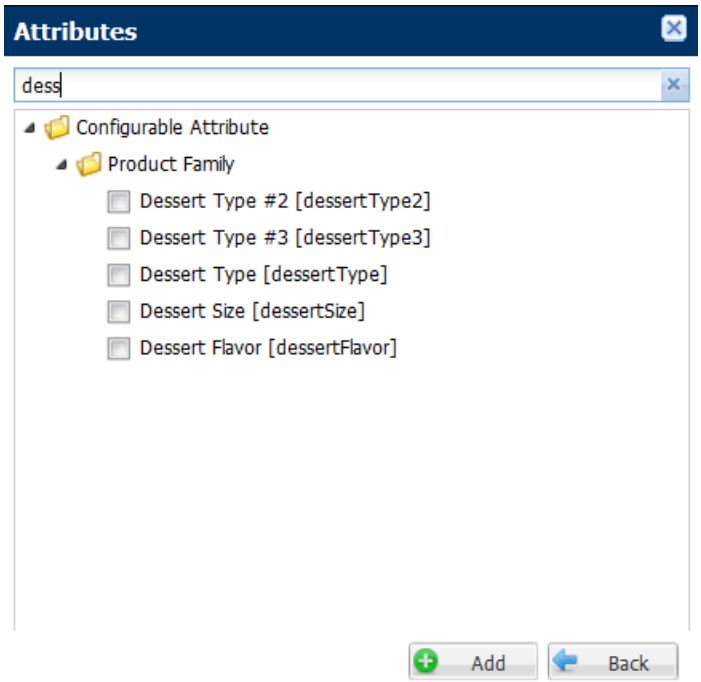
4. Choose the **Operator** from the drop-down menu.
5. Select the **Value Type**:
   - **Attribute:** Compare one attribute to another in a single condition row. *Attribute data type must be equal for the comparison. Also, when comparing an array attribute to a non-array attribute, the array attribute must be specified on the left side of the operator.*
   - **Static:** Enter a static value in the Values column.

Example

In the image, below, we've selected they array attribute, Dessert Type and did a simple attribute comparison to the non-array attribute, Dessert Type #2.



6. Set **Row Grouping**.
7. Select to apply the rule to **Configuration, Selection or Both**. *In this example, we've selected Configuration.*
8. Select an **Action Type**: Standard or Table Based. *In this example, we've selected Standard.*
9. Enter **Action Attributes** by clicking Add Attribute. *This launches the Action Attributes window.*
10. Select **configurable (array-type) attribute(s)** in this window and click **Add Attributes**. *In this example, we've selected array attributes Dessert Size and Dessert Flavor.*



11. Click on an individual attribute name. Notice that in doing so, the attribute values to constraint/allow, hide or set (based on rule type) will change as you select different attributes.
12. Set the values (these options change based on rule type):

Constraints (Values to Constrain/Allow)	Recommendations (Values to Set)	Hiding Rules (Values to Hide)
Constraint All: Select checkbox to constrain all values for the array attribute.	**Select Values**: Use drop-down menu to choose which array values to set.	Will hide attributes based on conditions.
Select Values: Use drop-down menu to choose which array values to constraint.	**Set Type**: Choose from None, Set and Force Set.	
Constrain Behavior: Choose from "Do not allow selected values" or "Only allow selected values"		

13. Create **Message to Display on User Side.**
14. Click **Save**.

NOTES



In the condition, you can use an attribute comparison.

When comparing an array attribute to a non-array attribute, the array attribute must be specified on the left side of the operator.



Adding array conditions to an existing rule will clear the advanced BML scripts of action attributes.



Convert to Advanced Condition is disabled when simple conditions contain an array attribute.

RELATED TOPICS

USE CASE - FIRE CONFIGURATION RULES ONCE PER SESSION

OVERVIEW

What Does It Do?

Some rules require gathering data from external sources and consequently, can be time consuming. When the data doesn't change rapidly, triggering the web service for every Configuration Update isn't necessary. In this article, we'll show you how to manage configuration rules so they will fire once, and only once, during a configuration sessions (both Add and Reconfigure mode).

How Does It Work?

For the Add Scenario

- Configurable attributes runOnceBegin and runOnceEnd each have their default value = true. The rules that we want to execute only once will have the entry condition as runOnceBegin == runOnceEnd. Making sure the rule only executes if both the flags match.
- Towards the end of the 1st iteration the flag runOnceEnd will be flipped, so that future update passes for the current configuration session the flags do not match and the corresponding rules don't run.
- Upon Add To Quote/Save, the value of runOnceEnd will be pushed to the commerce line level attribute cfgRunOnce_line.

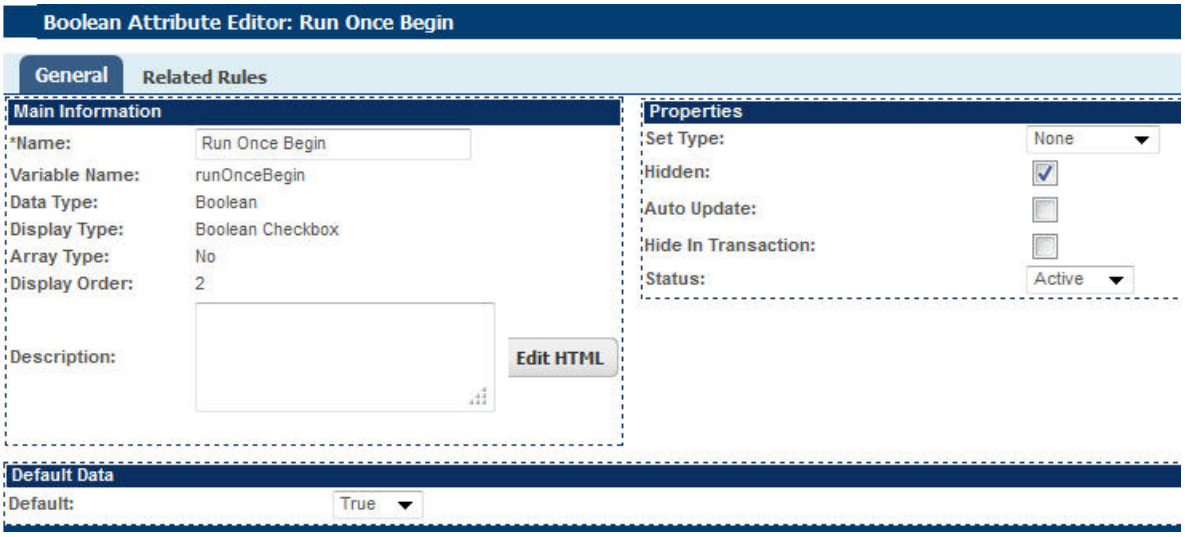
For Reconfigure Scenario

- Upon reconfiguration, the value of cfgRunOnce_line will be pulled and stored in the config attribute runOnceBegin. This will ensure that both the flags match for the 1st iteration.

ADMINISTRATION

Step 1: Create New Configuration Attributes

1. Navigate from Admin Home Page > Products > Catalog Definition > All Product Families > Configurable Attribute > List.
2. Click **Add**.
3. Name the attribute: **Run Once Begin (runOnceBegin)**.
4. Select the **Boolean** radio button.
5. Click **Add**.
6. Select **True** for the Default option under Default Data.
7. Select the check-box for the **Hidden** option.



8. Click **Update and New**.
9. **Repeat Steps 3-7**. Name the attribute: **Run Once End (runOnceEnd)**.
10. Click **Update**.

Step 2: Create New Commerce Attribute

1. Navigate from Admin Home Page > Commerce and Documents > Process Definition > *Commerce Process* > Documents > List > Line (Sub-Document)> Attributes > List.
2. Click **Add**.
3. Name the attribute: **Cfg Run Once (cfgRunOnce_line)**.
4. Choose **Boolean** as the Attribute Type.
5. Click **Add.**
6. Click the **Default Tab**.
7. Select the **Data** radio button.
8. Select the **True** radio button.
9. Click the **Document Views** tab.
10. Click **Select All** under **Hide** for all options.
11. Click Update.

Step 3: Line Item Default

1. Navigate to your commerce process > Documents > List > Line.
2. Click the Line sub-document link.
3. Verify or Select Define Advanced Default.
4. Copy the runOnceEnd to cfgRunOnce_line for new or reconfigured items. Click to see a sample script.

```
a. lineRet = lineRet + line._document_number + "~cfgRunOnce_line~" + getconfigattrvalue(line._document_number, "runOnceEnd") + "|";
```

5. Click **Update**.

Step 4: Configuration Layout Editor

1. Navigate from Admin Home Page > Products > Catalog Definition > Product Families > List > *Product Family* > Configuration Flows.
2. Click the **Layout** link next to a configuration flow.
3. Add attributes to the Layout Editor.
4. Click **Save**.

Step 5: Configuration Rules

1. Navigate from Admin Home Page > Products > Catalog Definition > Product Families > List > *Product Family* > Recommendations.
2. Click **Add.**
3. Create two rules: **SetRunOnceBegin(setRunOnceBegin)** and **SetRunOnceEnd (setRunOnceEnd)**.
   - These should have their conditions set to **Always True**.
   - Select the **action attributes** for which you want to run the rule only once.



All the configurable attributes for which we want to run the associated rules only ONCE per configuration session must be present between **runOnceBegin** and **runOnceEnd**. This is important because configuration rules run based on the order of the attributes.

4. Click **Add**.
5. Navigate to each configuration rule that you want to run only ONCE per session.

6. Add the condition: **runOnceBegin == runOnceEnd** to each.

7. Re-order attributes and rules to ensure that all attributes for which you want to run only ONCE per session are present between **runOnceBegin** and **runOnceEnd**.



This is important because configuration rules run based on the order of the attributes.

RELATED TOPICS

Article Submitted By: Savvis

CONFIGURATION PAGES - AN OVERVIEW

OVERVIEW

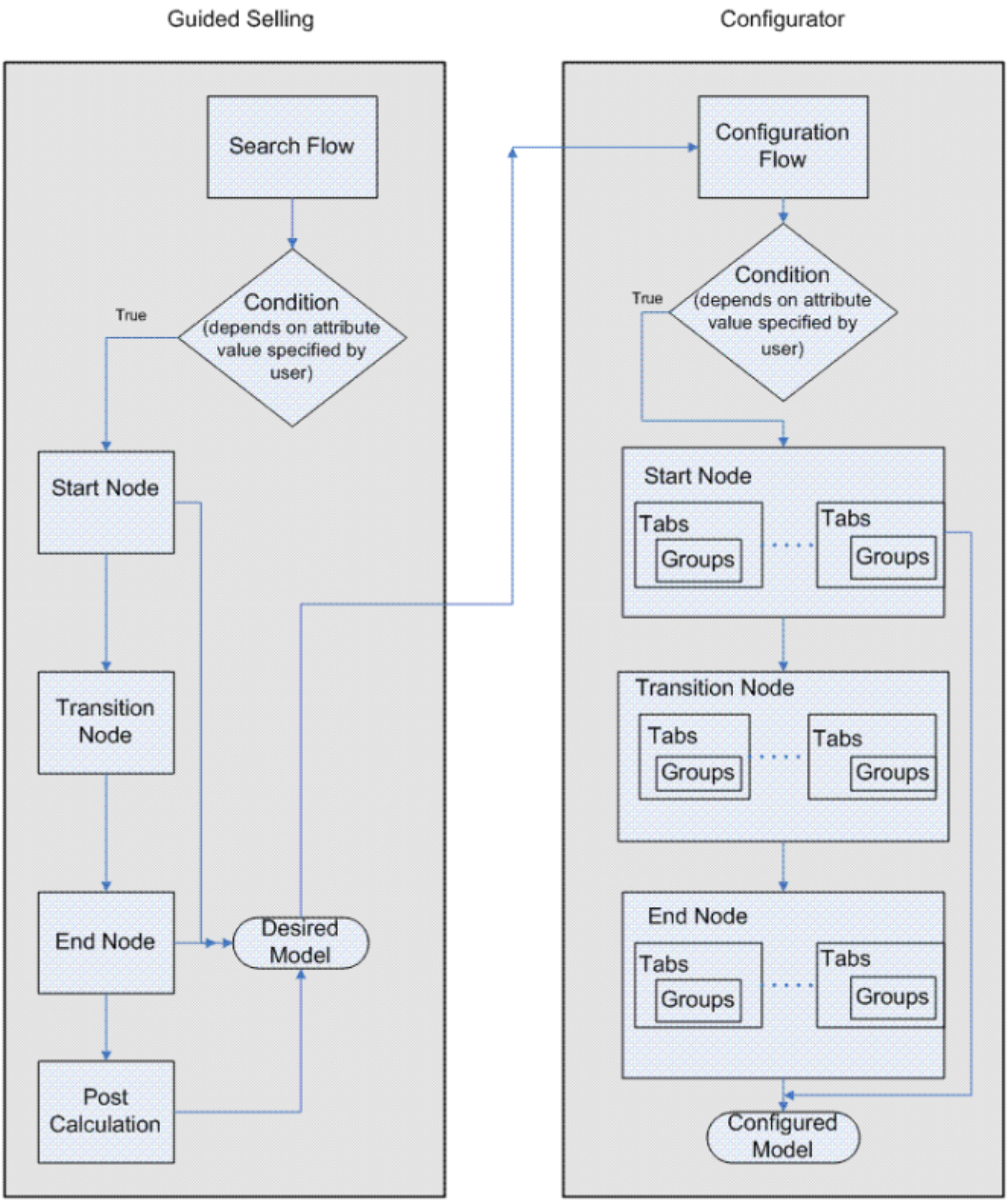
Flows are special rules that help guide buyers through the multi-step search and configuration process. *There are 2 types of flows:*

1. Configuration Flows
2. Search Flows

Configuration flows walk buyers through the product configuration process, enabling you to display configurable attributes in a logical sequence rather than all at once. A simple flow may contain a short process with only a single configuration step, whereas an advanced flow may contain a complex configuration process with many steps.

Search Flows, or selectors, enable you to create advanced search mechanisms for buyers. Within each segment, you can create multiple types of selectors to help buyers locate product lines, models across multiple product lines, or models within a single product line. A search flow gathers attribute values from the buyer and then uses those values to help buyers select suitable products. Search Flows are used for Guided Selling that helps target specific products with the desired configuration to the right buyer.

Page Workflow


Guided Selling
Search Flow
Condition
(depends on attribute value specified by user)
True
Start Node
Transition Node
End Node
Post Calculation
Desired Model
Configurator
Configuration Flow
Condition
(depends on attribute value specified by user)
True
Start Node
Tabs
Groups
Tabs
Groups
Transition Node
Tabs
Groups
Tabs
Groups
End Node
Tabs
Groups
Tabs
Groups
Configured Model

NOTES



If your system has configuration cache enabled, then all changes to flows data must be followed by a segment deploy in order to update the configuration cache.

RELATED TOPICS

CONFIGURATION PAGES - AN OVERVIEW

OVERVIEW

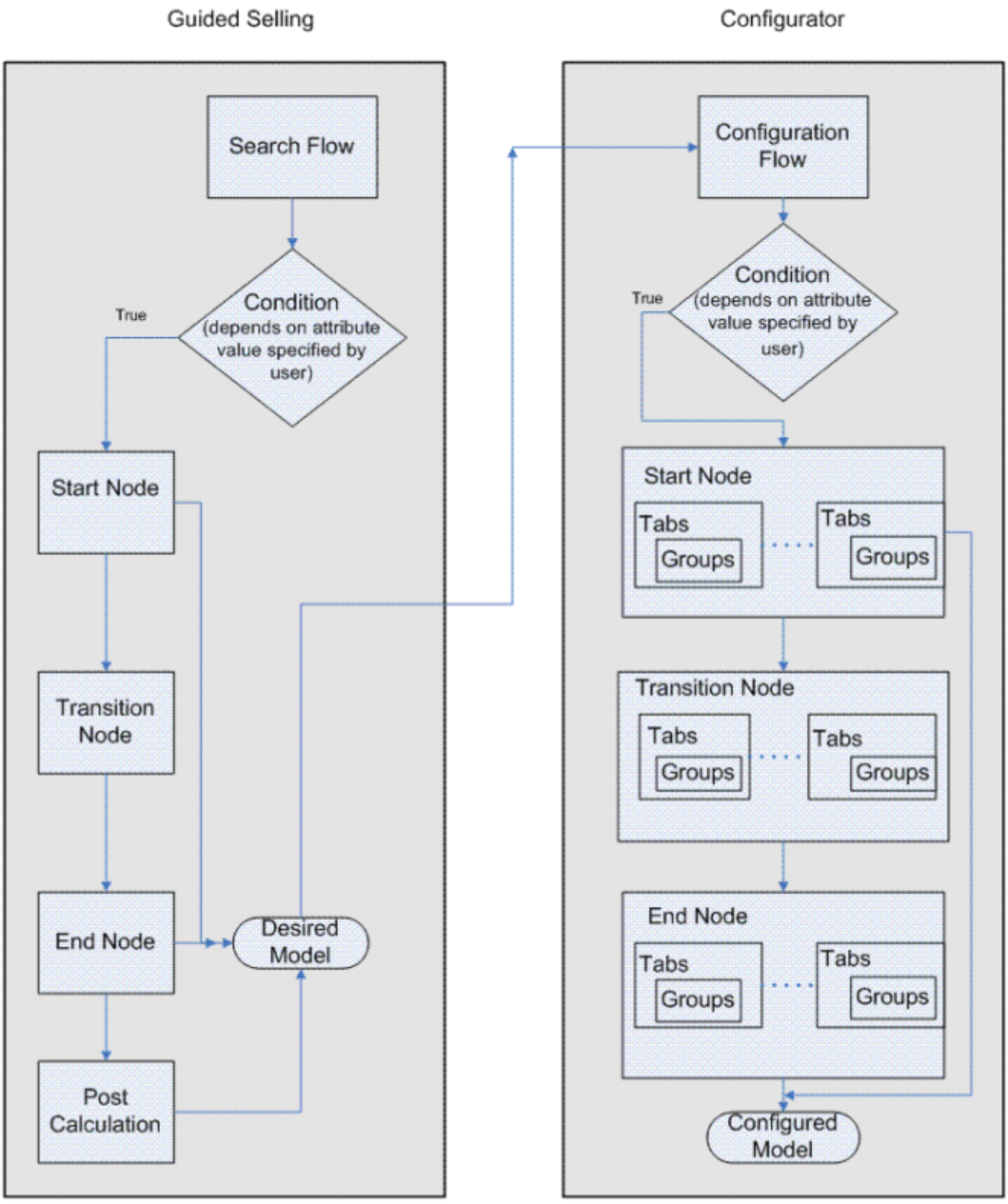
Flows are special rules that help guide buyers through the multi-step search and configuration process. *There are 2 types of flows:*

1. Configuration Flows
2. Search Flows

Configuration flows walk buyers through the product configuration process, enabling you to display configurable attributes in a logical sequence rather than all at once. A simple flow may contain a short process with only a single configuration step, whereas an advanced flow may contain a complex configuration process with many steps.

Search Flows, or selectors, enable you to create advanced search mechanisms for buyers. Within each segment, you can create multiple types of selectors to help buyers locate product lines, models across multiple product lines, or models within a single product line. A search flow gathers attribute values from the buyer and then uses those values to help buyers select suitable products. Search Flows are used for Guided Selling that helps target specific products with the desired configuration to the right buyer.

Page Workflow


Guided Selling
Search Flow
Condition
(depends on attribute value specified by user)
True
Start Node
Transition Node
End Node
Post Calculation
Desired Model
Configurator
Configuration Flow
Condition
(depends on attribute value specified by user)
True
Start Node
Tabs
Groups
Tabs
Groups
Transition Node
Tabs
Groups
Tabs
Groups
End Node
Tabs
Groups
Tabs
Groups
Configured Model

NOTES



If your system has configuration cache enabled, then all changes to flows data must be followed by a segment deploy in order to update the configuration cache.

RELATED TOPICS

CONFIGURATION FLOWS

OVERVIEW

What are they?

Configuration Flows are used to walk buyers through the configuration process. Configuration flows help arrange and organize configurable attributes to make configuration user-friendly. For products with large sets of configurable attributes, flows can be used to group different attributes and display them on separate tabs. Configuration flows can also be set up to direct users to different pages or flows based on their input selections.

How do they work?

A configuration flow rule consists of a condition and flow attributes. The flow is triggered when the condition is true. The condition usually depends on user specified values for displayed attributes (account, configurable, regular or user attributes).

Evaluation of a Configuration Rule

A status check is done to evaluate whether a flow rule is Active. The flow rule must be active for the condition to be evaluated. A flow is inactive if any of the condition attributes are not available to the current user. All the condition attributes must be available to the user in either the current or previous flows for the rule to be valid. An attribute is considered to be in a flow if it is either in the reference attribute, current display, affected attribute or next display attributes list.

Flow Selection Logic when Multiple Flows are Applicable

When more than one flow is applicable at the same time, the flows are sorted based on the following criteria (in sequence) to pick up the first flow:

- Descending order of validity of flows
- Descending order in product hierarchy (i.e.: model-level rules have highest precedence and all-family level rules have lowest precedence).
- Ascending order of node type. See **Rules for Selecting Node Types**.
- Descending order by number of rule inputs (Rules with simple inputs are evaluated last)
- Ascending order by rule order number (See Reordering Rules)

Rules for Selecting Node Types

1. The first flow must be one of the following node types: Start Node, Start and End Node, Primary Start Node or Primary Start and End Node. Otherwise, attributes are displayed using the default wizard.
2. If more than one flow is defined, then the first flow has to be one of the following: Start Node, Primary Start Node or Transition Node, and the last row has be of node type: End Node (NOTE_TYPE_START_END) for all the flows to show up.

Node Precedence

1. Primary Start Node
2. Primary Start and End Node
3. Start Node
4. Start and End Node
5. Transitional Node
6. End Node

Flow Selection for Different Cases

There are several different ways one can reach the model configuration page to run the flow selection logic. For each situation, the logic is given below:

Punch-in or Selector Search Result:

When the configuration page is reached through punch-in or selector search results page, the first flow condition s evaluated based on the selector input values. To be considered as a first flow for simple and advanced type condition, all the configurable condition attributes must be present in the selector input. Otherwise, the flow is not picked up.

Configure by Navigation Through Model Detail Page:

When the configuration page is reached through the model detail page with no selector input value, only the flows without configurable attributes as a condition attributes can be selected as the first flow. "Always True" type flow can be picked up as first, even if it has a configurable attribute as a condition.

Reconfigure from Commerce Document or Favorites Item:

A model can be reconfigured from a commerce document line item or from a commerce list item. The first flow is picked up based on the values for the configurable attributes that are stored. The flow that must be used next is re-evaluated based on the user's input selections.

Update Button on Model Configuration Page:

Configuration pages can be updated by either first flow or another flow. The first flow is always evaluated using the users input selections. Other flows are evaluated based on the attribute values read from the request, which could be either user's input, selector search results, or recommended values.

Next Button on Model Configuration Page:

When the next button is clicked, the next flow is evaluated as described in the selection flow logic from the list of flows with exception of flows that have been used so far.

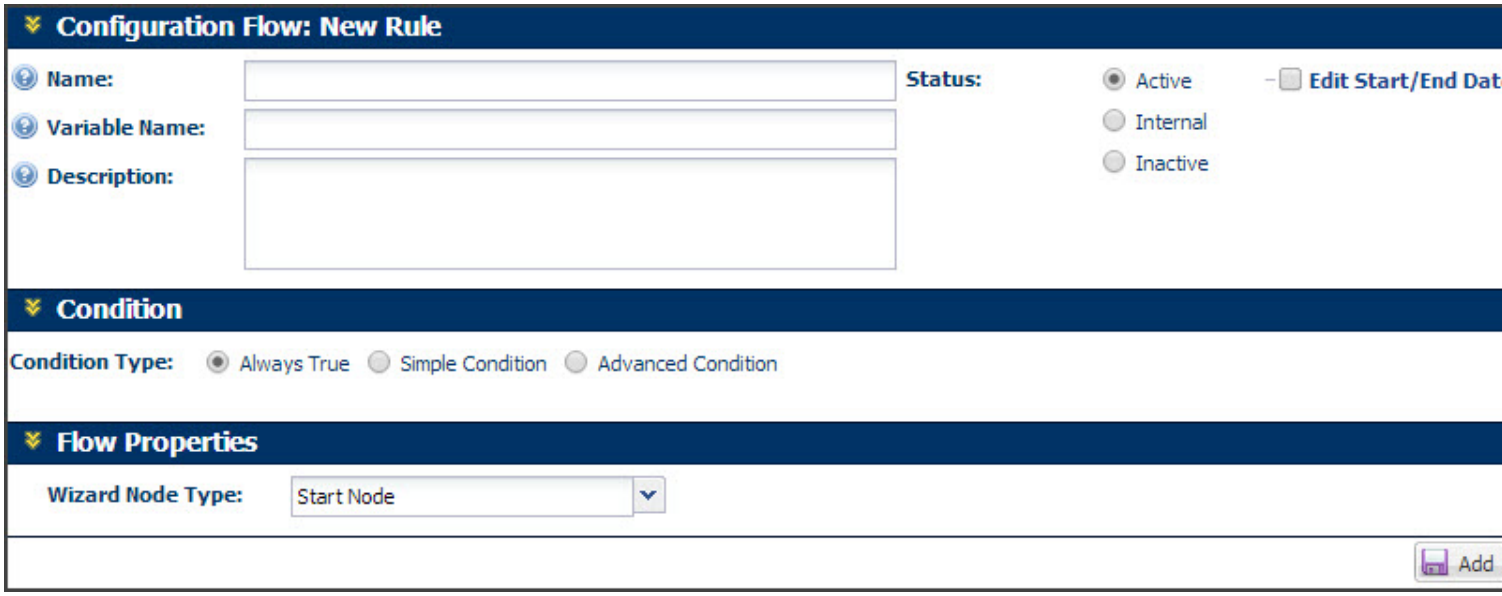
Flow selection through SOAP:

the flow selection logic works in the same way as in the case of UI. To select a particular flow as the first flow, all the condition attributes with the values satisfying the flow condition for the flow to be picked up is specified as selector input values in the SOAP input XML.

ADMINISTRATION

Creating a Configuration Flow

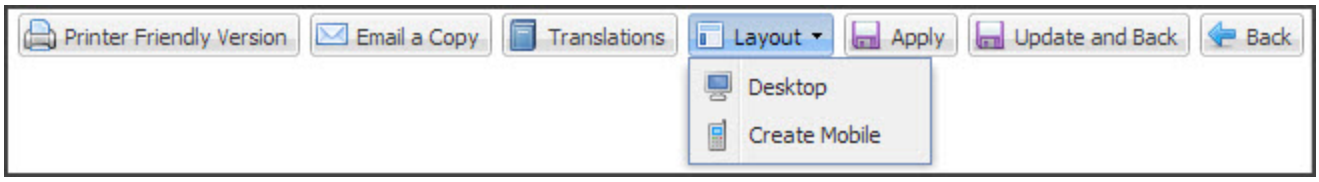
1. Navigate from Admin Home Page > Products > Catalog Definition > All Product Families > List > Select Product Family > Configuration Flow
2. Click **Add** to create a new flow rule.
3. Enter the **Name**, **Variable Name**, and **Description** of your rule.



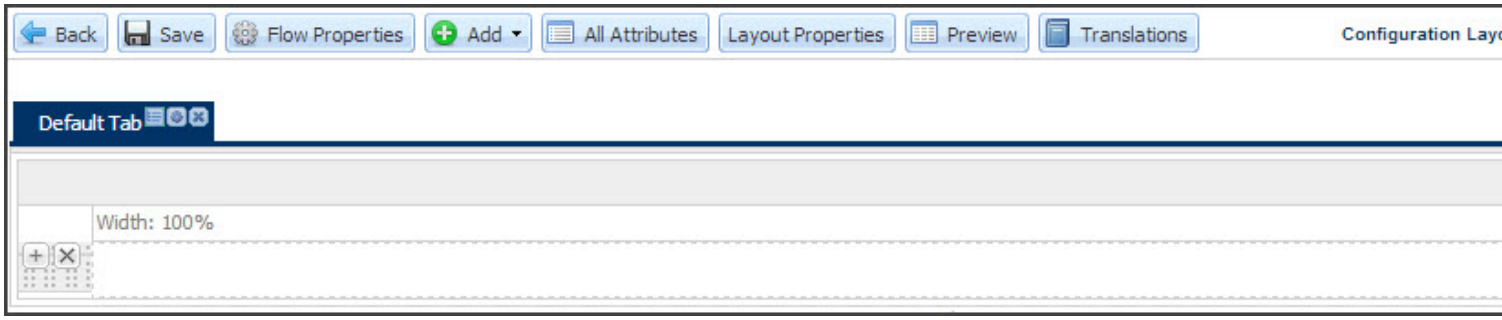
4. Select the **Status** of the rule.
   - **Active:** The rule must fire if the condition is satisfied.
   - **Inactive:** The rule does not fire, even if the condition evaluates to true.
   - **Internal:** Only fires if the FullAccessUser is the user and the condition evaluates to true.
5. If applicable, click **Edit Start/End Dates** to select the time period for which the rule is valid. *If no Start/End dates are selected, then it is assumed that the status is valid until the fullaccessuser changes it.*
6. Choose your **Condition Type.**
7. Specify the **Wizard Node Type:**
   - **Start Node:** Where the config flow will begin. You can have more than one start node, so if you plan on having multiple, this will be what you set your beginning nodes to.
   - **End Node:** Final node of the flow. This is where you'll see your action buttons like Add to Quote, as well as any recommended line items. As with start nodes, you can have multiple end nodes. The proper end node is set by attributes in the condition.

- **Start and End Node:** Start and End Nodes are combined.
- **Transition Node:** Transitions user from one node to another based on attribute selections.
- **Primary Start Node:** When there is more than one start flow, this lets the application know to start here first.
- **Primary Start and End Node:** Primary Start and End nodes are combined.

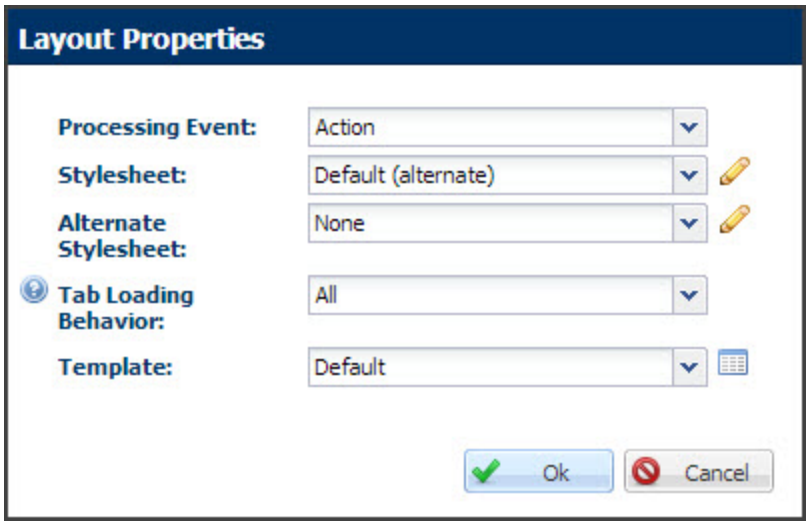
8. Click **Add**. *The following buttons will appear below the newly created Configuration Flow Rule:*



9. Click **Layout**.
10. Select **Desktop** or **Create Mobile** to edit **Layout Properties**. *For more information on creating a Mobile Configuration Flow Rule, see Mobile Configuration.*



11. Click **Layout Properties**. The following dialog will appear:



12. Specify the **Processing Event**:
    - **Action:** The condition for the flow is evaluated when an action occurs (the user clicks a button) on the Model Configuration page.
    - **Value Change:** The condition for the flow is evaluated when the users change a value for a configurable attribute.
    - **Tab Change:** The condition for the flow is evaluated when the users move from one tab to another on the Model Configuration page.
13. Select a **Stylesheet** from the drop-down menu. Click the pencil icon to make changes to the stylesheet.
14. Select an **Alternate Stylesheet** from the drop-down menu. Click  to make changes to the stylesheet.
15. Select **Tab Loading Behavior**:
    - **Load all tabs on page load:** Loads all data across tabs.
    - **Load only active tab on page load:** Loads data on only the active tab in configuration. Note: Users might see a loading page appear when the switch tabs, as the next set of data is being loaded. Select this option improves performance and rendering speed on larger configuration flows.
16. Select a **Template.** Click  to preview the template.
17. Click **Ok**. *This will close the **Layout Properties** dialog.*
18. Click **Save**.
19. Click **Add** to begin setting up the user-side layout.



Desktop and Mobile Layouts are considered separate Flows. If a change is made to one flow, it will not automatically be reflected in the other. User must make changes in BOTH the Desktop and Mobile Layout.

NOTES



All attributes are considered to be in the default flow.

The flow node type must be Primary Start and End Node or Start and End Node for multiple flow cases in SOAP API. This is because multiple flow functionality is not supported in this case.

For a forced end type flow, all attributes are considered in the flow.

If due to setup error, the condition attributes appear in the forward flows (next flows), the rule is marked as invalid.

If there is an End Node flow at the model level and a Start Node at the product line level, End Node flow will be picked up because of the ordering of the product hierarchy.

RELATED TOPICS

SEARCH FLOWS

OVERVIEW

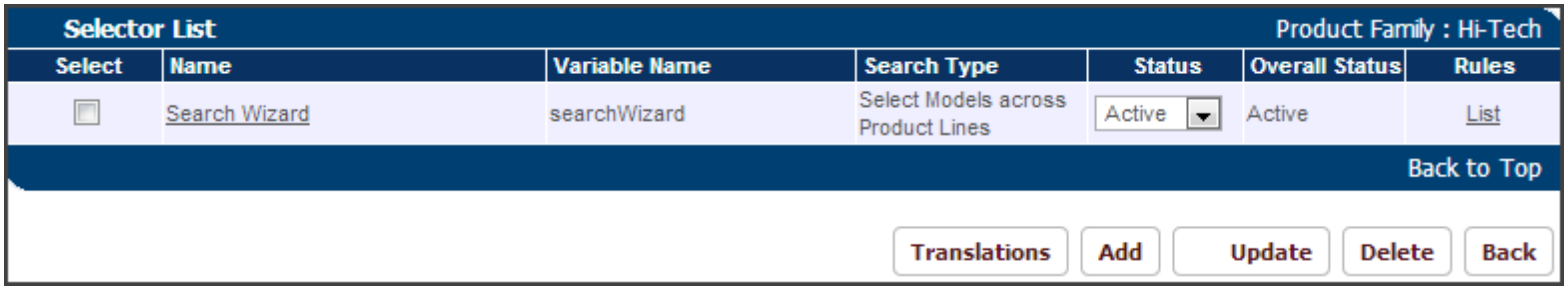
The primary purpose of Search Flows is to help buyers find appropriate products. Suppliers benefit from creating search flows because they know buyers have assistance in locating complex products in a less error-prone way than with manual selection. Buyers benefit from search flows because they can find the best products for their needs quickly and accurately. Search flows usually appear to buyers when they click on the Advanced Search button in the Search page.

A search flow typically contains selection attributes whose values can be used to determine a set of products. Selection attributes are configurable attributes that can be displayed to the buyer in steps as a flow or all at once. Using flows, you can add logic to the search process by displaying ”next” steps based on previously set values. With some extra planning, search flow attribute values can carry over to the product configuration process. **In order for this to work, 2 conditions must exist:**

1. The attribute name and type (e.g. float, integer, non-numeric) in the search flow must match up exactly with the corresponding attribute used in the configuration process.

2. The attributes whose values should pass along to the configuration process must be included in the search flow's post-calculation.

A search flow rule has a condition and an action. The values of the condition attributes determine the result of the condition, which when true triggers an action. The action is to pick up the right search flow to be displayed on the selector configuration page. If more than one search flow rules exist and have their condition evaluating to true, then the first flow is selected based on the flow selection logic mentioned below.



Flow Selection Logic when Multiple Flows are Applicable

When more than one flow is applicable at the same time, the flows are sorted based on the following criteria (in sequence) to pick up the first flow:

1. Descending order of validity of flows

2. Ascending order of nodeType.

3. Descending order by number of rule inputs (Rules with single inputs are evaluated last)

4. Ascending order by rule order number (See Reordering Rules

Following is the precedence order for flow nodeType:

1. Start Node

2. End Node

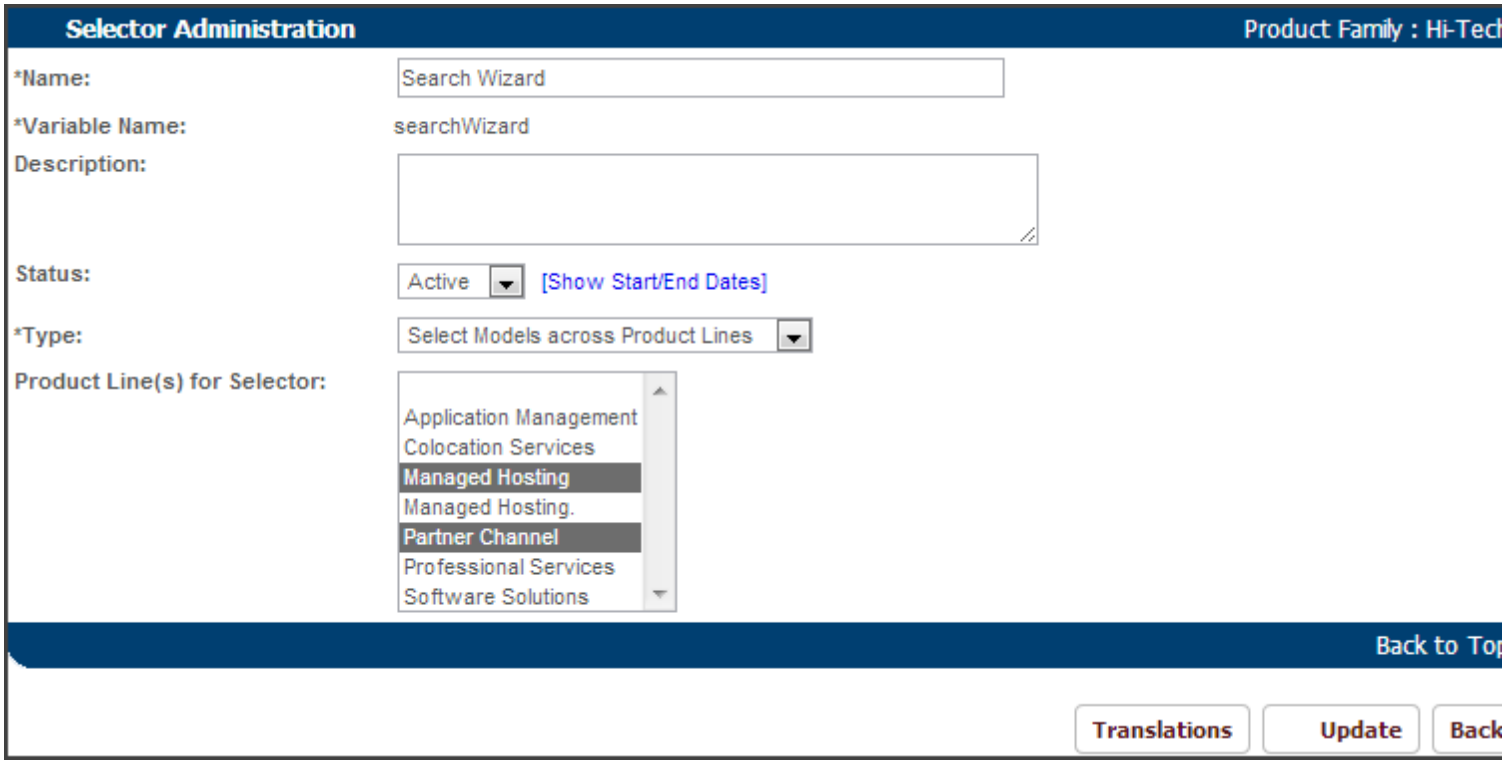
3. Start and End Node
4. Transitional Node
5. Primary Start Node
6. Primary Start and End Node

The rule for selecting node types for flows is that, the first flow has to be one of first four node types listed above for the flow to be picked up and showed on the model configuration page, otherwise the attributes are displayed using the default wizard. If there are more than one flows defined, then the first flow has to be any one of first three nodes listed and the last flow has to be of type NODE_ TYPE_END for all the flows to show up.

ADMINISTRATION

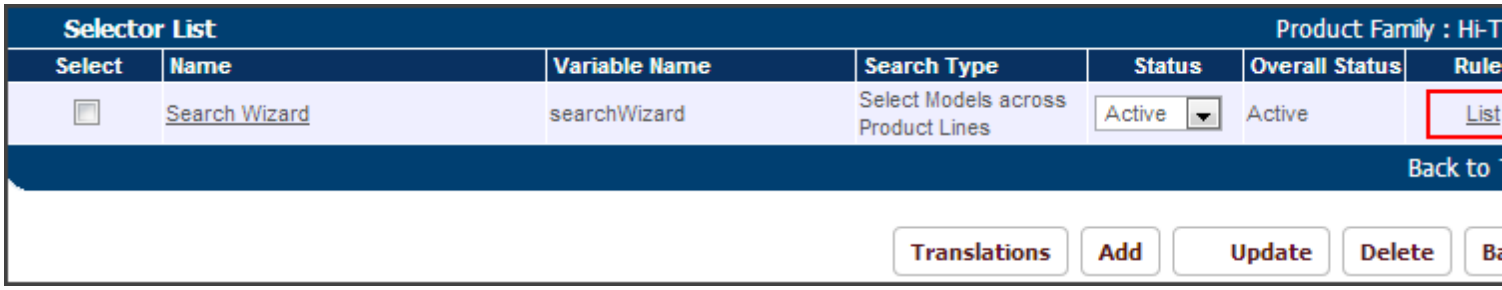
Creating a Search Flow

1. Navigate from: Admin Home Page > Products > Catalog Definition > Product Families< Search Flow (for specific family).
2. Click **Add**.
3. Specify the Name, Variable Name, Description and Status of the rule.
4. Specify the **Selector Type**. There are three options:
    - **Select Models in Single Product Line:** A search flow created for this selector helps the buyer choose the optimal models in a specified product line.
    - **Select Product Lines**: This helps the buyer choose the optimal product line int hat particular family.
    - **Select Models Across Product Lines:** Helps the buyer choose the optimal models within the specified product family.
5. Select Product Lines for the Selector(s).

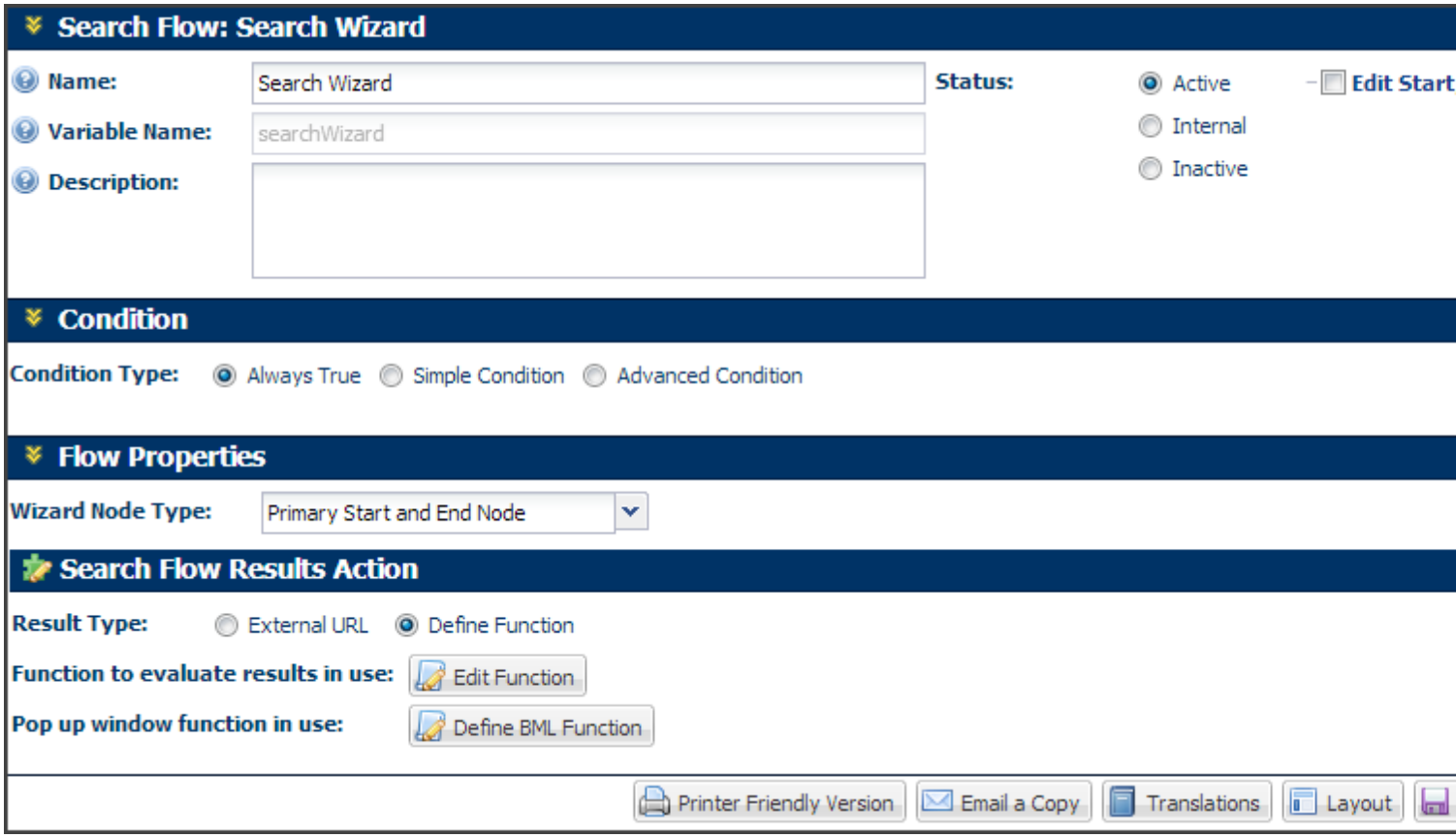


6. Click **Add.**

7. Click **List** under the Rules header.



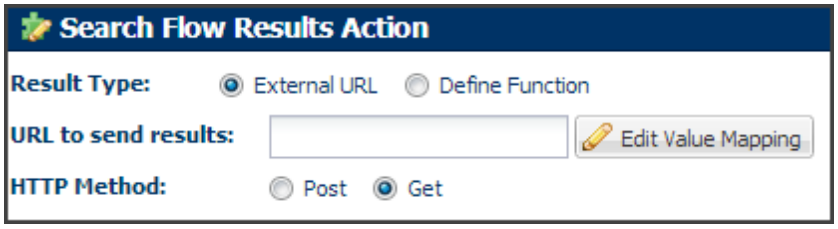
8. Click **Add**.



9. Enter the **Name**, **Variable Name** and **Description** of your rule.
10. Select the **Status** of the rule:
    - **Active:** The rule must fire if the condition is satisfied.
    - **Inactive:** The rule does not fire, even if the condition evaluates to true.
    - **Internal:** Only first if the fullaccessuser is the user and the condition evaluates to true.
11. Select the **Start/End Dates** for which the rule is valid (if applicable).
12. Select the Conditions for the rule.
13. Select the **Wizard Node Type** from the drop-down menu.

14. Select the **Result Type:**

    - **External URL:**



    - **Define Function:** See Selector Pop-Up: Define Function (Search Flow Results Action) section.

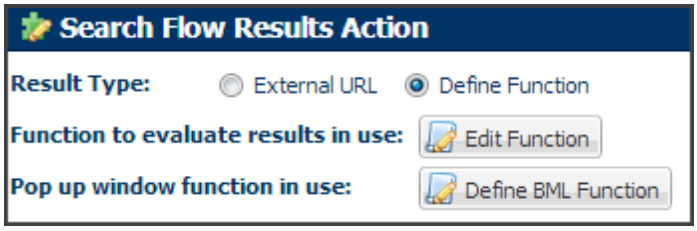
15. Click **Update and Back**.

> Search Flows can only be created at the Product Family level.

Selector Pop-Up: Define Function (Search Flow Results Action)

Selector Pop-ups provide a way to improve the performance of the Search Results page. They introduce the ability to invoke calculations on a need-by basis on the Search Results page. *A link/button is displayed to the user to launch a pop-up dialog with the desired calculations. The calculations can be tailored to individual model/product line(s) returned as search results. The calculations can access dynamic values on the search results page.*

1. Navigate to the **Search Flow Rule Editor** page. *See above steps.*
2. Choose **Define Function** as the Result Type.



3. Click **Edit Function** next to **Function to evaluate results in use**.
4. Define the function and include a call to the pop-up function. Click here to see syntax.

```
<?xml version="1.0" encoding="UTF-8"?>
<Item>
      <Company hidden="yes">testCompany</Company>
```

```
      <Description>
      <GroupName title="Product Line"r;>bigWidgets</GroupName>
      <Name hidden="no" title="Model"r;>W1</Name>
      </Description>
      <Attributes>
      <Attribute title="SIZE"r;>40</Attribute>
      <Attribute title="r;Color">RED</Attribute>
      <Attribute>
            <![CDATA[
            <table cellspacing=0 cellpadding=0>
                  <tr>
            <td class=button-left><img src="/img/button10.gif" width=10 height=22></td>
            <td class=button-middle>
      <a class=button-text href="javascript:popupFunction()>Calculator</a>
            </td>
                  <td class=button-right><img src="/img/button10.gif" width=10 height-
                  t=22></td>
            </tr></table>]]>
      </Attribute>
      </Attributes>
</Item>
```

5. Inputs to the popupFunction:
    - All the attributes available to the Function to Evaluate Result Set are available to the popupFunction
    - Dynamic values present in the search results page can be passed as a parameter to the popupFunction
6. Output of the popupFunction
    - The pop-up Function returns a string
7. Syntax of popupFunction
    - popupFunction (search_results_string) where,search_results_string: This is an optional string parameter that is used to pass values computed in the search results function to the popupFunction. It enables a fullaccessuser to pass dynamic values on the search results page. The maximum size of this string in 1000.

NOTES



If there is an End Node flow at model level and a Start Node at the product line level, the End Node at the model level will be picked up first since the refType is higher up in the sequence.



Search Flows can only be created at the Product Family level.

Selectors can have multiple flow nodes.

Recommendations, Constraints, and Hiding Rules can take effect in Selectors.

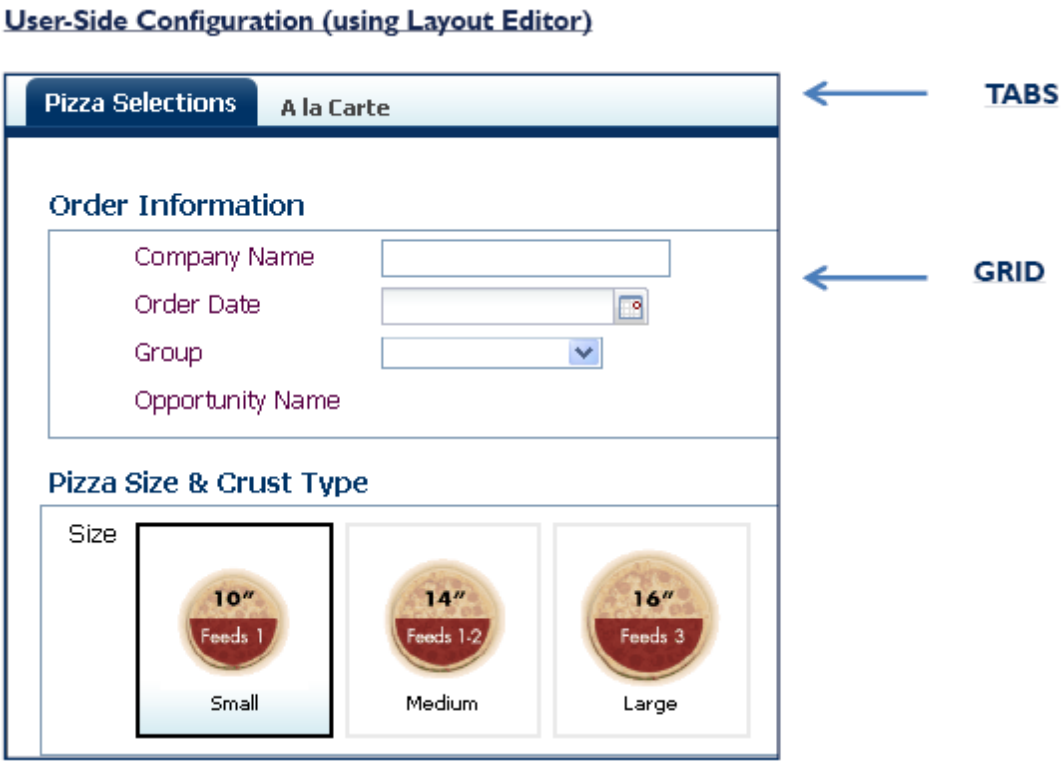
RELATED TOPICS

CONFIGURATION LAYOUT EDITOR

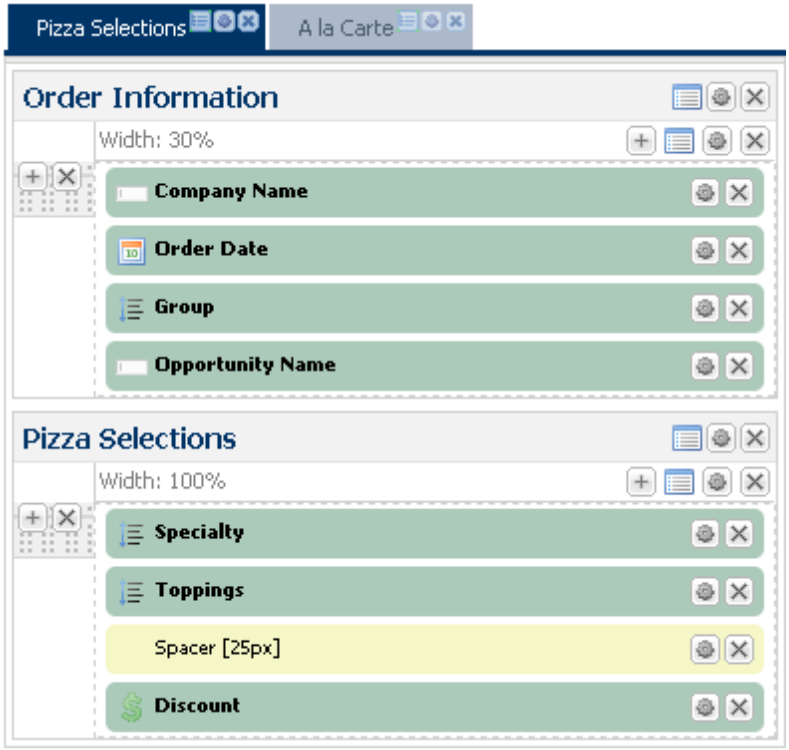
OVERVIEW

You can create easily create standard configuration layouts. At the core of this functionality, the Configuration Layout Editor provides a graphical drag & drop feature for creating and managing the display of the model configuration page(s). You have the ability to create multiple grids (formerly known as groups) and tabs, greater flexibility in laying out UI elements and more, from a single page. Once you have created your configuration flow, you can open the Configuration Layout Editor and begin to layout the user-side configuration page. You can access the Layout Editor from the Configuration Flow List page (displayed as a link) or within a configuration flow (displayed as a button).

User-Side Example



Configuration Layout Page Example



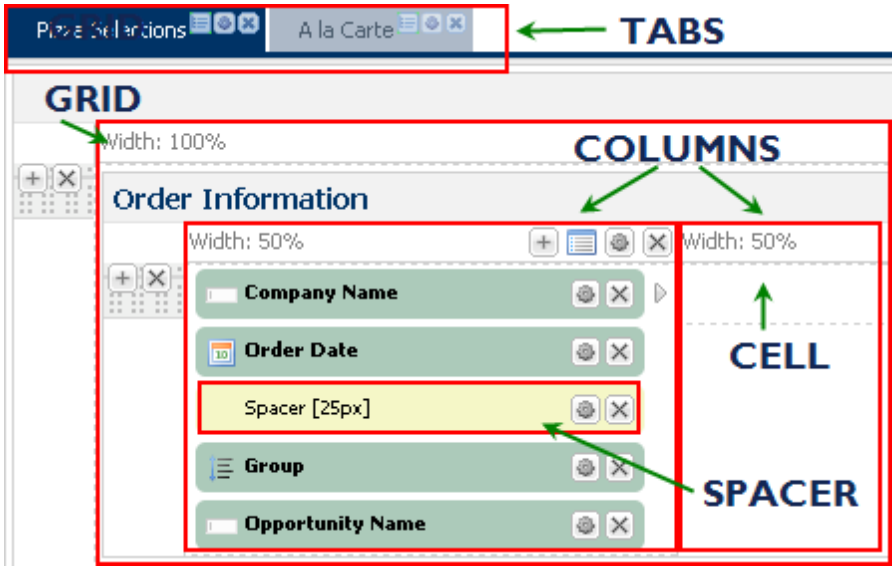
ADMINISTRATION

You can customize and brand the user-side configuration UI in an easy and intuitive way thanks to drag & drop functionality, previews, grouping and more. The Layout Editor allows you to customize the look and feel of your configuration UI by using Tabs, Grids, Flow and Attribute Properties and a Configuration Stylesheet.

Visual Editing Area

The Visual Editing Area is a fixed-width page that is divided into the following areas:

- **Visual Editing Area:** This is for the visual editing of the Commerce UI Layout
- **Action Toolbar:** These are actions that can be performed on the visual editing area.



Action Toolbar

The Configuration Layout Editor contains a strip of administrative action buttons.

Action Button	What does it do?
Back	Returns the user to the Configuration Flow Administration Page. The application does not save if this button is used. **Note: You will lose your work without warning! Make sure to save!!**
Save	Saves the layout to the database. An asterisk will appear next to the Save label any time a change is made in the visual editing area to remind you to save.
Preview	Opens a preview window of the user-side interface based on the latest changes to the configuration page layout editor.
Flow Properties	Returns you to the Configuration Flow Properties page where you defined your flow. Note that the variable name is now read-only.
Add	Opens a menu to allow you to add either Attributes or Layout Elements. When clicked, these options display respective menus.
All Attributes	Opens a window where you can edit the display properties associated with any attribute that is referenced on this configuration page layout (note: this includes all tabs on the current layout). Filtering features help you to find the specific attribute or group of attributes that your are looking for.

Icons Legend

Icon	Description
	Click to add a row or column element to a grid.
	Click to delete a layout element; tabs, grids, columns, rows, and attributes.
	Click to edit element properties such as tab properties, grid properties, and column properties.
	Click to edit attribute properties like labels, displaying help and set type.

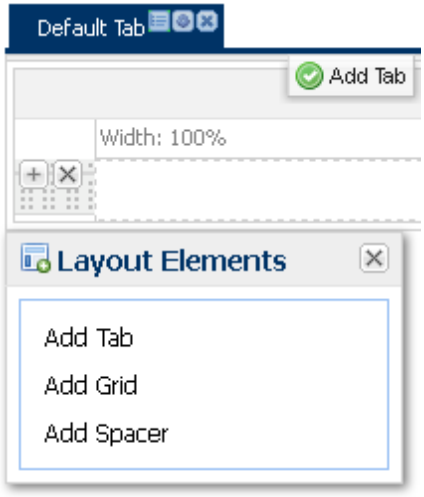
Adding Layout Elements

The configuration layout editor allows the fullaccessuser to add elements to the page and nest some page elements within each other to achieve a pleasing user experience. Some layout elements are color-coded for faster identification.

Setting up Tabs

The first step in organizing your configuration page(s) will be to create tabs. You can add as many configuration tabs as you'd like. When you first open the configuration layout editor, you'll notice that one tab is already available: Default Tab. Rename this tab and use it, or add additional tabs and delete this one later.

1. Click **Add** from the Admin Action Bar.
2. Select the **Layout Elements** menu.
3. Drag and Drop **Add Tab** from the Layout Elements menu to the top bar.



A green check mark lets you know that the cursor position where you'd like to place the tab is accepted. A pop-up window will appear where you define the Tab Properties.

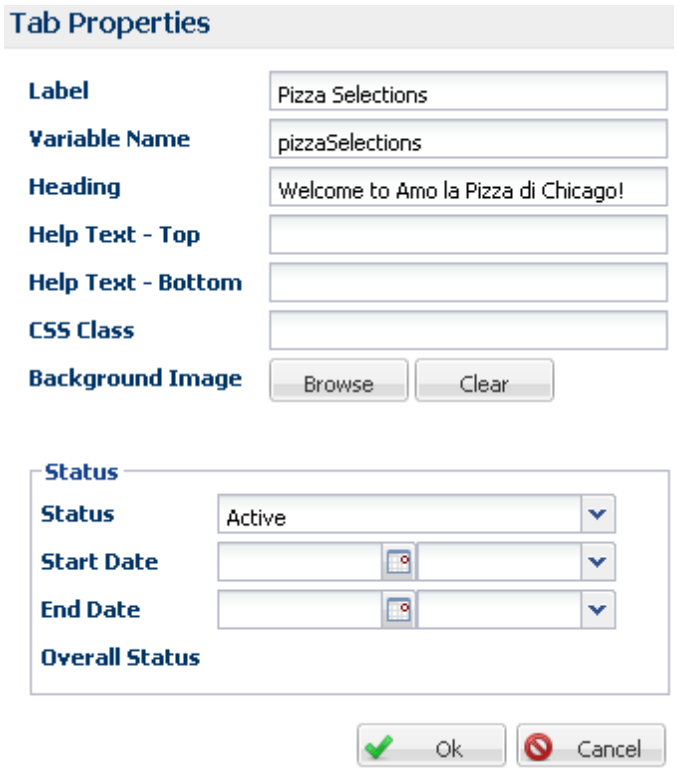
4. Set the **Tab Properties**.

The tab properties can also be found by clicking on the icon next to the tab name.

5. Enter a unique **Label and Variable Name** for your Tab. You will not be able to change the variable name of the tab. These fields are REQUIRED.

All other fields in the Tab Properties window are optional. You are not required to enter values, though it allows for further individualization of your application.

6. Select one or more of the following options:
   - **Heading:** This applies the provided text to the top of the Tab page. Tab page heading that appears directly under the tab name on the user side. This will not display in the visual editing area.
   - **Help Text:** This displays help text either below the tab page heading or at the bottom of the tab page.
   - **CSS Class:** When populated, this setting applies the specified text as a class name to the tab element. Note: make sure to add this class and its properties to the stylesheet applied to the configuration flow.
   - **Background Image:** Click Browse and select an image from the File Manager.
   - **Tab Status and Dates:** If "Inactive" is specified, the tab will be inactive overall. If "Active" is specified with no date range, the tab will be active overall. If "Active" is specified with a date range, the overall status will be Active only during the date range; outside the date range, the overall status will read "Inactive". "Internal" status works the same with regard to the date range and overall status as does "Active".



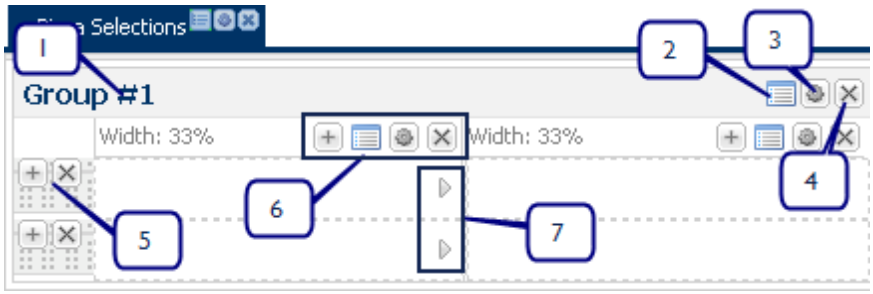
7. Click **OK** to save change or click Cancel to close the window without saving changes.

The current tab will be shown in blue and the others will be displayed in gray.

The default tab also contains a single default grid. By default, one tab, one grid, one row and one column are required for a layout and cannot be deleted.

Adding Grids and Grid Elements

Grids are used to organize layout elements, attributes and array sets. This functionality replaces Groups from previous versions of BigMachines. There is a default grid provided within the Default Tab. You can create a new grid and delete the default grid later, if you desire.



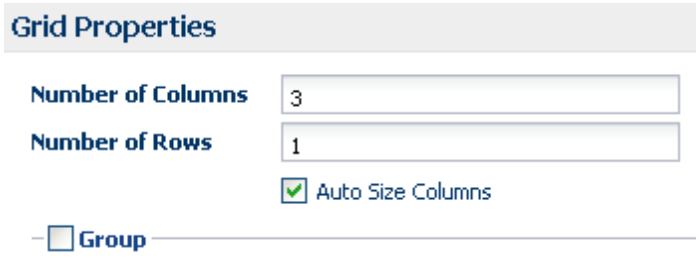
Position	Element	Description
1	Grid Header	The grid header identifies each grid. You can drag and drop grids by clicking on the header. Grids can be nested, meaning you can drag them inside other grids. **Note**:if only one grid element spans the entire horizontal width of the layout, this grid element may not be dragged and dropped elsewhere. When a title is applied to the grid, the text in the grid header will change to this title, currently "Group #1".
2	All Attribute Properties	Opens a window where you can edit the display properties associated with any attribute that is referenced inside the current grid element. Note: this includes all tabs on the current layout.
3	Grid Properties	Opens a window where you can edit grid properties such as the number of columns and rows, group title, CSS class, alignment and more.
4	Grid Delete Button	Deletes a grid and grid elements, including columns, rows and attributes.
5	Row Add/Delete Button	Click the + button to add a row and click the X button to delete a row. Note: Be careful when deleting a row as it will delete all contents within the row as well.
6	Column Button Bar	Click the + button to add a column and click the X button to delete a column. Use the gear icon to adjust the column width. The list icon will open the All Attributes window.
7	Column Span Arrows	Use the arrows to eliminate one or more column boundaries by spanning across columns. You can return the column to an unspanned state.

If you have just created a tab, the Layout Elements window should still be open. If this is the case, you can skip Steps 1-2.

1. Click **Add** from the Admin Action Bar.
2. Select the **Layout Elements** menu.
3. Drag and Drop **Add Grid** from the Layout Elements menu to the top bar. A green check mark lets you know that the cursor position where you'd like to place the grid is accepted.
4. Enter the **number of columns and rows** you'd like within your grid, when the the Grid Properties window appears.
5. Select the Auto-size Columns box if the columns should be auto-sized.

> Auto-Size will set the Column Layout to 100% and divide the size equally among the columns. If this is left un-checked, you will change the column widths using the Column Properties button (gear icon) within each column.

6. Select the **Group checkbox** if you'd like to set-up your Grid like a Group. This will open up additional group properties. You can have up to 200 rows.



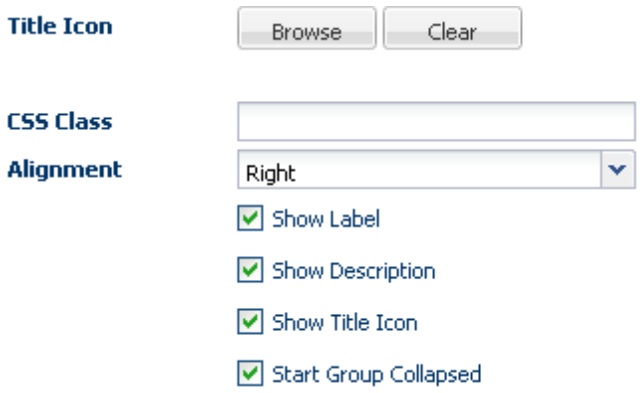
7. Enter a **Group Label**. This will appear in the Grid Heading. You can also add a description to let other fullaccessusers know what this group is for.



8. Click **Browse** to select a Title Icon from the File Manager. This will appear to the left of the group label. Select Clear to remove images.
9. Enter a **CSS Class for the Group**.

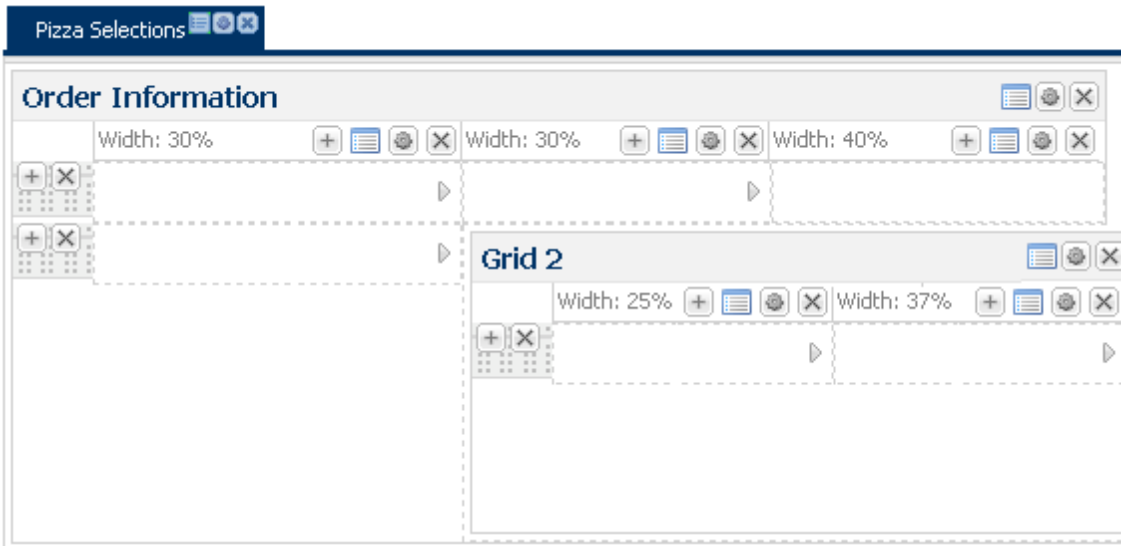
When populated, it applies the specified class name to the DIV that surrounds all elements in the grid, allowing you to style one group differently from another.

10. Select the **Alignment** of the group label: Left, Center or Right.
11. Select checkboxes for the following options:
    - **Show Label:** When checked, the Group Label will be displayed on the UI with the label alignment you selected in Step 10.
    - **Show Description:** When checked, a link containing the group description will appear on the user-side.
    - **Show Title Icon:** When checked, the icon that you selected in Step 8 will appear on the user-side.
    - **Start Collapsed:** When checked, the grid will start collapsed when the user first comes to the page.



12. Click **OK** to save your changes or click Cancel to close the Grid Properties window without saving.
13. Click **Preview** to take a look at how your changes affect the user-side layout. Remember to click SAVE if you'd like to Save what you've done.

    Grid nesting, the practice of placing one grid inside another, gives you ultimate layout flexibility.



If you add a row into a grid that contains a nested grid, the new row will be added beneath the nested grid.

You can drag and drop existing grid elements to another column or row. All contents of the grid will be transferred to the designated location, including specific column widths. Once in the new location, the column widths will expand or contract to the new element in which it resides.

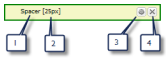
If you specify a number of pixels, please ensure that the layout element(s) that contain the attribute have sufficient width to accommodate the pixel width you specified. If sufficient width is not available, unsightly visual shearing will occur on the user-side interface. Alternately, when setting a percent width, understand that the numeric input is the percent of the total width

White space is preserved for a cell, even when all its attributes are hidden, when adjacent cells in the row have visible attributes.

White space will collapse within a cell when an attribute is hidden. It will also collapse for an entire row when all attributes within the row are hidden.

Adding Spacers

A Spacer is just a blank space with a defined height. Spacers are displayed vertically between other inputs. The default height is 25px. In the Visual Editing Area of the Layout Editor, a Spacer should be displayed with the following:



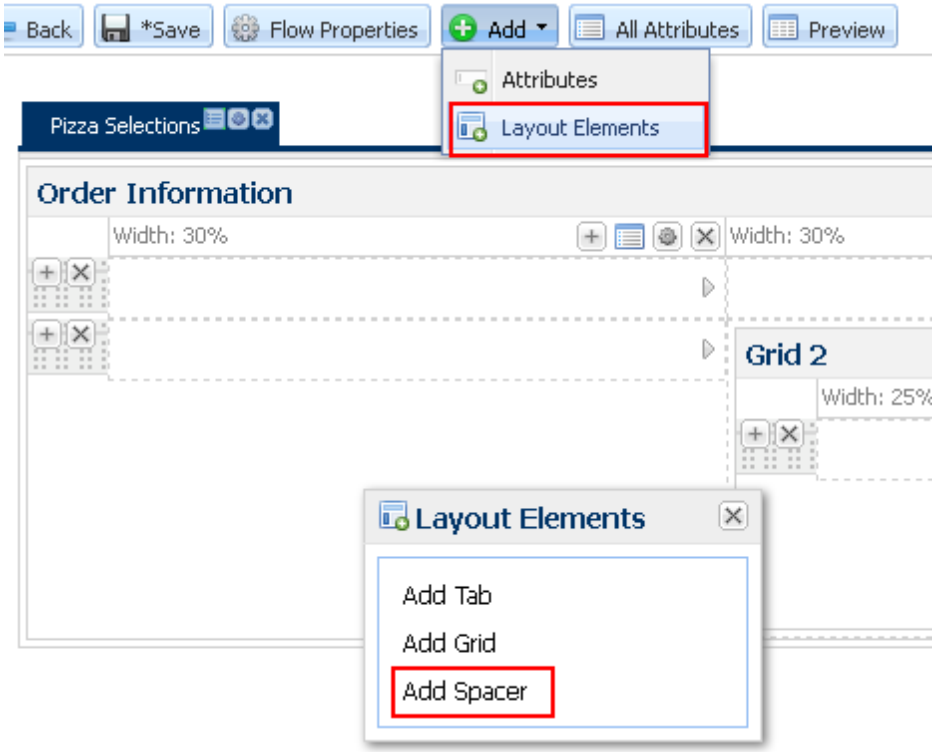
1= Static Label: Spacer

2= Read-only display of spacer's height value

3= Gear icon to open Edit Properties window & edit the height (in pixels)

4= Delete icon to remove the spacer from the column

1. Click **Add** in the Administration Actions toolbar.
2. Click **Layout Element** from the menu.
3. Select **Add Spacer** from the menu.



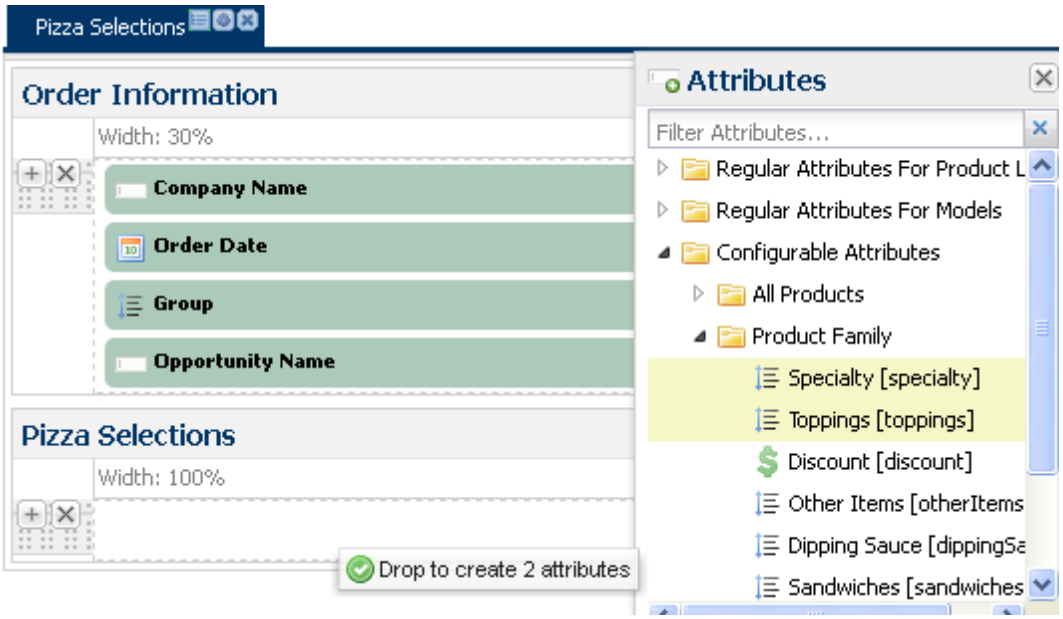
4. **Drag and drop** the Spacer into a cell. The red dotted line represents where the spacer will be placed. The green check mark indicates it can be dropped in that space.
5. Determine the **height** of the Spacer in pixels.
6. Click **OK** to save changes or click Cancel to return to the Layout Editor without saving the changes.

To edit a spacer, click the gear icon to the right of the label and change the height pixels.

If a spacer is added into a cell with an attribute and the attribute is hidden, the cell will not collapse.

Adding Attributes

1. Click **Add** in the actions toolbar.
2. Choose Attributes from the drop-down menu. Drag and drop one ore more attributes into the editor using SHIFT + click to select more than one attribute from the Attributes Menu.
3. Drag and drop the selected attribute(s) into a cell. The red dotted line represents where the attribute will be placed. The green check-mark indicates that it can be dropped in that space.



4. Edit any attribute in the layout by clicking the gear icon on the far right side of the attribute to open up the Attribute Properties window.

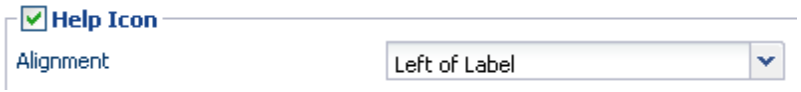


From here, you can edit only the current attribute's properties and behaviors. Follow the steps to edit attribute properties, starting from the Attribute Properties window.

5. Choose and edit properties for the **Display Label**, on the Layout Tab, in order to show the attribute label name. De-selecting this will hide the attribute label to the user. It will also not allow you to set additional related properties.
    - **Separator**: Determine what you'd like to use to separate the label from the attribute value. What is entered here will display immediately after the label.
    - **Alignment:** Determine the alignment of the label: Left, Center or Right.
    - **Width:** Enter the label width in pixels or percentages.

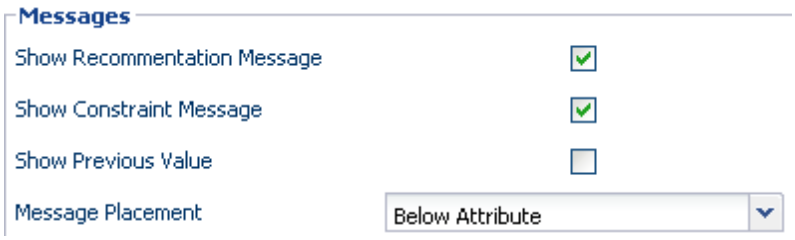


6. Enter the **properties** for the attribute value fields.
    - **Position:** Determine if you'd like to display the attribute value field to the right of the label or below the label. The default is right of the label.
    - **Width:** Determine the field width in pixels or as a percentage. Select Auto to let the application determine the width.
    - **Alignment:** Determine if you'd like the field to be left, center or right-aligned.
7. Select the **Help Icon checkbox** if you'd like to show a help icon. If selected, an option will appear allowing you to choose where to display the icon, to the left of the label or to the right of the attribute value.



8. Select additional options:
    - **Hide Lock**: Select to hide the lock next to attributes. *This is typically used to modify recommendation behavior.*
    - **Show Calculator:**Select to show the attribute calculator icon next to the attribute.

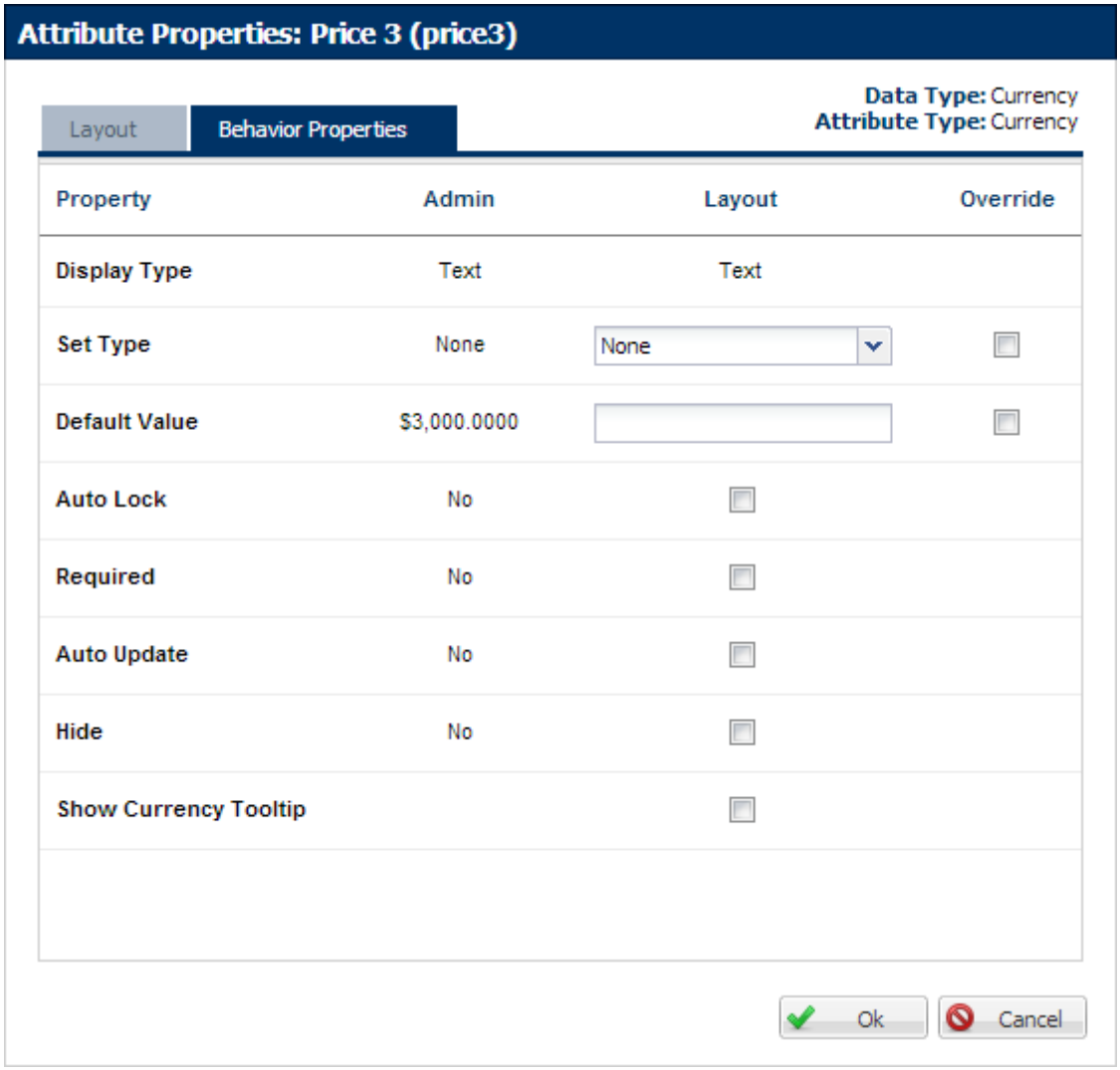
9. Select **Message** options:
    - **Show Recommendation Message:** Select to show messages when Recommendations are active on the attribute.
    - **Show Constraint Message:** Select to show messages when Constraints are active on the attribute.
    - **Show Previous Value:**Select to show the previous value of the recommended or constrained attribute. This will appear when an update occurs, either by clicking an Update action or if the constraint rule is marked as an AJAX.
    - **Message Placement:** Use the drop-down menu to select the **placement of the messages**. You can display messages below the attribute, right of the attribute value, bottom of the group, bottom of the page and the top of the page.



10. Click on the **Behavior Properties** tab.



On this tab, you can manage the different display settings of the attribute. You can also override the display type, set type and default value properties that were defined when you created the attribute.



The options available in the Layout column vary according to the data type of the current attribute. For instance, this string text field's Display Type single select menu contains the options: Text and Text Area. However, if this were a single select attribute, the Display Type options would be Single Select Menu, Radio Button Horizontal, and Radio Button Vertical. Similarly, the Default Value options for a single select attribute would appear as a single select menu of the options that are available for the current attribute.

Note that the help icon allows the user to observe rich text help that is defined on that specific attribute's administration page. If an fullaccessuser has not defined help text for an attribute, the end user will

observe no result when they attempt to access the help icon.

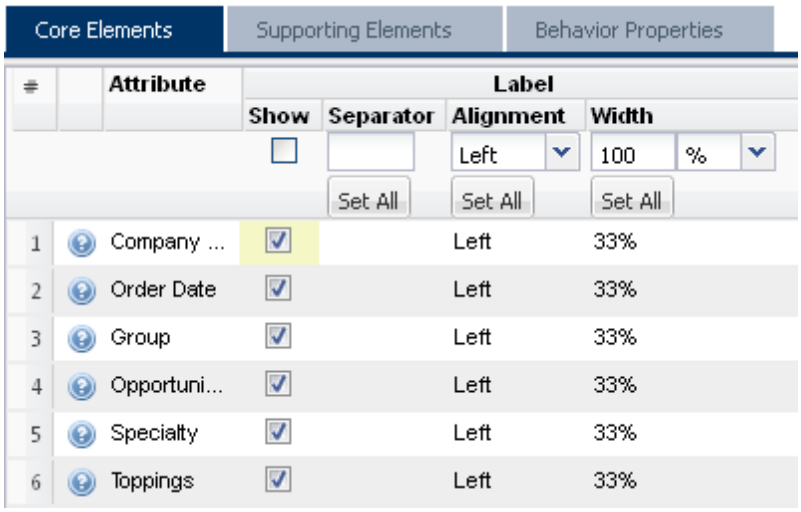
Managing Attribute Properties

Attribute Properties can be accessed from multiple locations on the configuration page. You can do so by clicking the All Attributes button in the Administration Action Bar or you can click one of the List icons at the grid or column level.

In the Attribute Properties modal window, there are three tabs available to make changes to attributes: Core Elements, Supporting Elements and Behavior Properties. The purpose of this section is to allow you to make mass changes to attributes within the entire configuration layout, a grid or a column.

Core Elements

Within the Core Elements tab, you can make changes to the display of the Attribute Label, Value or Message. You can also select whether individual attributes are auto-locked and if they should display the recommendation message, constraint message and the previous value.

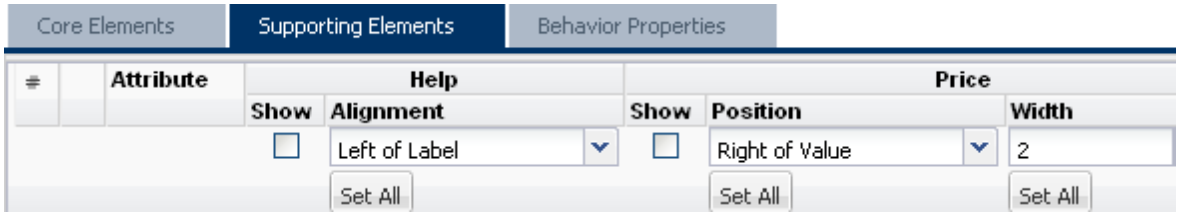


In the image above, you see the name of each attribute names. Corresponding to each attribute, you may elect to show or hide that attribute's label on the user interface. In this administration interface, you may affect the displays of many attributes at once. You can adjust settings, attribute by attribute or, if you wish to set a given display element en mass, input the appropriate display element in the top-most input row. If a "Set All" button is provided, clicking it will apply the setting you entered to all attributes listed below.

Example: If you wish to set the label width to 100% for all attributes in the list, clicking the "Set All" button in that column will apply "100%" to all attributes listed below .It works in a similar manner for Value and Message display settings.

Supporting Elements

Within the Supporting Elements tab, you can make changes to display of the Help Icon, the position and width of Price (Attribute Value Pricing), the position of the Menu Image and whether or not to show or hide locks and calculators.



Behavior Properties

On this tab, you can manage the different display settings of the attributes. You can also override the display type, set type and default value properties that were defined when you created the attribute. This is also where you can determine, for each attribute, if they should be set to auto-update, required or hidden.

Click the Attribute Properties button at the grid or column level to make changes to ONLY the attributes within the grid and/or column. If you would like to make changes to all of the attributes within the layout, click the All Attributes button at the top of the page.

Conversion of Pre-12 Layouts

Pre-12 standard configuration layouts was composed of tab layouts and group layout. On upgrade, these layouts will be automatically converted to the latest configuration layout functionality.

Converting legacy tab layouts and group layouts

The column width of the container grid is calculated based on how many tab columns were present in the legacy layout. For example, a legacy two column layout is converted to a parent grid with two columns; each with a width of 50%. Similarly, a legacy one column layout is converted to a parent grid with one column of width 100%.

Legacy group layouts are converted to grid within the parent column. Each grid has as many columns as the old group. For example, a 3-column group will be converted to a 3 column grid. The number of attributes in the column determines the number of rows in the grid.

Column widths for grids are calculated by adding the legacy label and value widths. If the new grid column width is less than 100, then it is not scaled.

- Attribute label and value widths are now calculated by scaling proportionately to the new grid column.
- Attribute label and value widths within the new column add up to 100 %.

For example, consider a legacy one-column group where the label and width values are 20% and 40% respectively. This legacy group will be converted as follows:

- New grid with one column. The column width is 60% (20 + 40 – sum of old label and value layouts).
- Each attribute is in its own row. Therefore, this grid has as many rows as the number of attributes in the column.

- The new label and value widths for attributes are proportionately calculated as 33% and 67% respectively using the formula: (legacy width value/new column width) * 100.
- Attribute label and value widths add up to 100% in the converted layout.



If the width of groups was more than 100 % in the old layout, the new layout normalizes it to 100%.

Converting * and pixel values

- If a legacy group's columns were all defined in pixel or percentage, they are transferred in pixel or percentage widths.
- For legacy groups that mixed percentage and pixel widths, pixel is first converted to percentage [1280 pixel = 100%]
- For legacy groups that mixed pixel and *, the widths are first converted to percentage. Similarly, for groups that mixed * and percentage, all widths are first converted to percentage
- Asterisk conversion uses the formula: (100 - (sum of all percentage values)) / (number of asterisks)

Attributes in the layout

Each attribute is migrated in its own row in the new grid. It is strongly advised that you test the end user pages after upgrade to ensure that the page displays properly. If not, changing the widths of the columns should fix most positioning and width related issues.

Notes on Arrays within Layouts

Array Widths

Widths can be specified for labels, value fields and messages for array sets. Additionally, widths can be specified as auto-size.

1. Horizontal and Tabbed Array – widths specified in percentage
    - Label width defines the width of the label column.
    - Value width defines the width of the attribute field.
2. Vertical Array – widths specified in percentage
    - Lable width does not apply.
    - Value width defines the width of the attribute fields.

3. Pixel Widths – for all array set layouts
    - The attribute label and the attribute values widths are the widths specified in pixels.
    - If the attribute value width is wider than the column width, then a scrollbar is displayed.

Array Messages

1. The fullaccessuser can choose to display array messages in different ways - Messages can be displayed in five ways for each attribute – Below Attribute, Right of Value, Bottom of Group, Bottom of Page and Top of Page.
2. Additionally, it is now possible to set percentage or pixel widths for array messages.
3. Message display, look and feel have been greatly enhanced.

Miscellaneous

- Array indices can be removed from any specified position without losing the values specified in other positions.
- Array icons have been improved. Existing alternate Stylesheets must be edited to point to these new icons.

AJAX Rules

- Selecting a constrained value hides pricing, recommended items and invocation buttons automatically. Additionally, these elements reappear via AJAX when the constraint is removed.

Enhanced Look and Feel

- Groups can be collapsed by default.
- The fullaccessuser can specify one stylesheet and an alternate stylesheet for each layout.
- Help, images and group descriptions are rendered inline instead of as a popup.

TROUBLESHOOTING

Common layout issues after upgrade

Fields are cut-off

The new layout will cut-off fields if there is not enough space to display them within the column. You must update column widths after upgrade to display fields properly.

Styles not rendering properly

- Missing Stylesheets - Layout now supports only one stylesheet and an alternate stylesheet. For legacy layouts with multiple stylesheets, only the style sheet from the first tab will be associated with the layout. It should be noted that all legacy

stylesheets are available in the Stylesheet section. The fullaccessuser is advised to download those stylesheets and then combine them into one alternate stylesheet. Additionally, tab class names can be used as tab identifiers in this combined alternate stylesheet.

- Existing custom browser-specific styles may not work correctly anymore due to the DOCTYPE change – These custom styling are most likely no longer needed.
- HTML attributes not rendering properly because of the updated configuration stylesheet – Explicitly set styles for HTML attributes in the CSS.

Advanced Templates

All tabs using advanced templates are now in iframes where old html, old css, and old JavaScript are still in use. However:

- Any JavaScript in header/footer that accesses attributes in tabs that use advanced templates will no longer work (but it is possible to modify it to look for the attribute within the iframe)
- Any JavaScript inside an HTML attribute within a tab using advanced templates that looks for attributes in other tabs will no longer work (but it is possible to modify it to look for attributes in the parent document)

JavaScript customization does not work:

- Any previous custom JavaScript that relies on DOM relationships (e.g. .parent(), .children()) or on specific tag names will no longer work as expected.
- JS might break if it does not wait for DOM ready events.

Old JavaScript *(will break on upgrade)*

```
<script>
if (document.configurationForm && _BM_USER_LOGIN != "Creed Bratton" && _BM_USER_LOGIN != "Michael Scott" && _BM_USER_
LOGIN!= "Dwight Schrute") {
elements = document.getElementsByName("create_quote");
elements[0].parentNode.parentNode.parentNode.parentNode.parentNode.parentNode.style.display = "none";
elements[1].parentNode.parentNode.parentNode.parentNode.parentNode.parentNode.style.display = "none";
}
if (document.configurationForm){
elements = document.getElementsByName("start_over");
elements[0].parentNode.parentNode.parentNode.parentNode.parentNode.parentNode.style.display = "none";
elements[1].parentNode.parentNode.parentNode.parentNode.parentNode.parentNode.style.display = "none";
}
</script>


Sample Replacement JavaScript
<script>
```

Old JavaScript *(will break on upgrade)*

```
if (document.configurationForm) {
$('.button-start-over').hide();
$('.button-invoke-create').toggle(_BM_USER_LOGIN == "Creed Bratton" || _BM_USER_LOGIN == "Michael Scott" || _BM_USER_
LOGIN == "Dwight Schrute");
}
</script>
```

NOTES



Column widths add up to be 100 %.

Most styling should be done using the regular stylesheet UI. The alternate stylesheet can be used for additional CSS customization.Update existing alternate Stylesheets to use the latest array icons. Define CSS classes for tabs and groups to aid in advanced styling.



Widths specified for layout elements and attributes are respected. The fullaccessuser must ensure that enough space is available within a column to display attributes properly.

All attributes are migrated in their own row to maintain legacy behavior of preserving white space on hiding attributes. It is recommended that the fullaccessuser move attributes to a single cell if collapsed space behavior is desired. A single cell is considered the same column and same row on the layout.



Make sure you click SAVE before you leave any area of the Configuration Layout or your changes will not be saved.

RELATED TOPICS

CONFIGURATION PAGE TEMPLATES

OVERVIEW

Fullaccessusers can define templates that can be associated with configuration flows. This feature allows flexible display of pricing/recommended items sections on the configuration page and new default templates that can give your configuration a new look and feel.

ADMINISTRATION

Page templates are defined at the All Product Family level and can be associated with configuration flows at all levels of the product hierarchy, however page templates are not supported for Search Flows.

Predefined Templates

There are 3 templates available by default:

- **Default Template:** This template features the main content on top, the pricing section in the middle and the Recommended Items section at the bottom of the page. It is most similar in layout with the layout of the configuration page pre-11.
- **Right Side Bar:** This template places the pricing and the Recommended Item sections on a right bar beside the main content.
- **Left Side Bar:** This template places the Pricing and the Recommended Item sections on a left bar beside the main content.

The left and the right bar templates display Pricing and Recommended Items beside the main content. While these templates allow the you to showcase Pricing and Recommended Items, always verify that the content of the model configuration page looks good with these templates.

TROUBLESHOOTING

The look and feel of the templates will be compromised if:

- There are a lot of columns displayed for Recommended Items that induces a horizontal scroll.
- Too many Recommended Items and/or Pricing induce a vertical scroll.
- Size of the main content is out of proportion to the size of the Recommended Items and/or Pricing sections.
- If there are no Pricing and Recommended Items rules defined for the model, these sections will show up as blank on the Model Configuration page. The fullaccessuser is advised to use the Default Template in this case.

NOTES



If a different template is selected, deployment is needed.



For cached product families, changes to the template take effect on the end user side when that product family is deployed.

By default, pre-defined templates will be available for use in configuration flows only and they cannot be edited.



You cannot create custom configuration templates.

RELATED TOPICS

GUIDED SELLING

OVERVIEW

A search flow, or selector, offers more granularity and flexibility in navigating the product hierarchy, prior to entering the configuration flow. The selector can guide users to the correct model when they do not know what they are looking to configure in advance. For example, this would be necessary if the users do not know the products or if there are complicated calculations required to select the model.

ADMINISTRATION

Step 1: Creating a Search Flow

Please click here for information on Creating a Search Flow.

Step 2: Set Up a One-Node Flow

1. Click **List** under the Rules header for your Search Flow.



Selectors can have multiple flow nodes, just like configuration. This is useful when the fields for the user change significantly based on some input. Recommendations, constraints, and hiding rules can also take effect in selectors.

2. Click here to see steps for creating a Search Flow Rule.

3. Select Define Function as the Result Type.



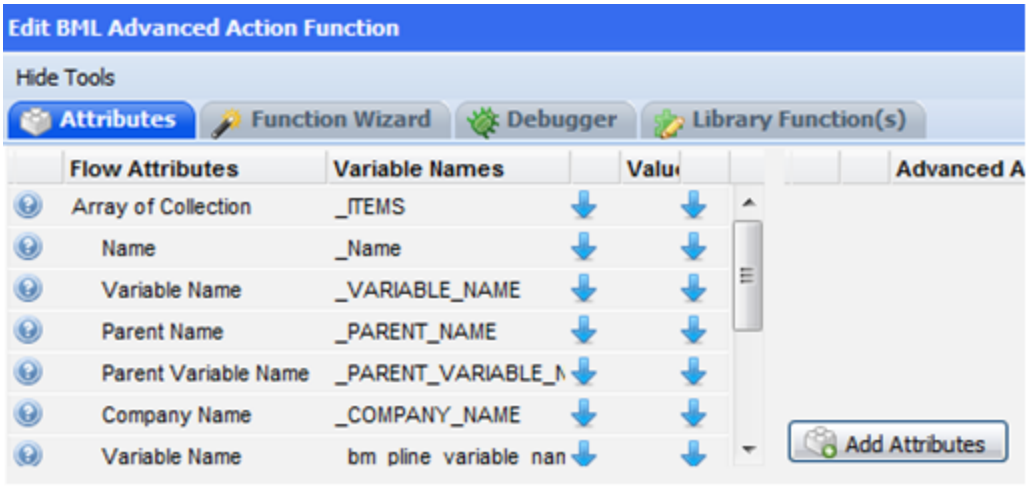
This script determines which model punch-ins will appear to the user, based on any condition attributes that you have added to the search flow.

4. In this example, Legacy Type and Product Line are single select menus. Their values (New, Old and X, S, NL) will determine which models appear, depending on whether the model is New or Old and whether it belongs to the X series, S series or NL series. *The BML script automatically imports the catalog attributes as parameters. Not all of these may be relevant for your use case.*

5. There are a few special attributes of the XML tags:

   - **button='yes':** For product line or model, show as a button. 'No' will show as a link.
   - **configuration='yes':** For model, destination of the button or link will go to configuration. If 'no', it will go to the model profile page.

- **configuration='yes'** :For model, destination of the button or link will go to configuration. If 'no', it will go to the model profile page.
- **sort='yes':** This attribute is sortable on the selector results page for the end user.
- **numeric= 'yes':** Define the sort behavior as numeric, so values are 1.0, 2.0 10.0 rather than string sort of 1.0, 10.0, 2.0.



6. The BML script returns a string containing XML. The XML has to conform to a specific hierarchy. Click here for an example.

```
<SelectorSearchResults>
    <Columns>
        <Column ... />
        <Column ... />
        ...
    </Columns>
    <Items>
        <Item>
            <Attributes>
                <Attribute ...>attributeText</Attribute>
                <Attribute ...>attributeText</Attribute>
                ...
            </Attributes>
            <Description>
                <GroupName hidden="yes" button="no">item._PARENT_VARIABLE_NAME</Grou
                <Name button="Yes" title="item._NAME" configuration="yes" align="cen
            </Description>
        </Item>
    </Items>
</SelectorSearchResults>
```

7. Click here to see the full script used in this example, with comments.

```
str = "<SelectorSearchResults>";

//Columns: These are the columns that show up in the selector search results
str = str +"<Columns>";
str = str +"<Column title=\"Product Line\" align=\"center\" width=\"25%\" sort=\"tru
str = str +"<Column title=\"Legacy Type\" align=\"center\" width=\"25%\" sort=\"true
str = str +"<Column title=\"Model\" align=\"center\" width=\"25%\" sort=\"true\">3</
str = str +"</Columns>";

modelDict = dict("string");

//A data table that maps the model names available for the search flow's search para
modelMap = bmql("select modelName from ModelMap where productLine = $productLine_sel
for model in modelMap {
    put(modelDict, get(model, "modelName"), get(model, "modelName"));
}

itemStr = "<Items>";
for item in _items { //_items is a list of all Models across the product lines for t
    if (containskey(modelDict, item._NAME)) { //If modelDict contains this specific
        itemStr= itemStr+ "<Item>";

        itemStr = itemStr + "<Attributes>";
        itemStr = itemStr + "<Attribute title=\"Product Line\" variableName=\"produc
        itemStr = itemStr + "<Attribute title=\"Model\" variableName=\"legacyType\"
        itemStr = itemStr + "</Attributes>";

        //This column needs to have the hidden GroupName node or the Model buttons w
        itemStr = itemStr + "<Description>";
        itemStr = itemStr + "<GroupName hidden=\"yes\" button=\"no\">"+item._PARENT_
        itemStr = itemStr + "<Name button=\"Yes\" title=\"" + item._NAME + "\" confi
        itemStr = itemStr + "</Description>";

        itemStr = itemStr + "</Item>";
    }
}
itemStr= itemStr +"</Items>";
if (itemStr <> "<Items></Items>") {
    str = str + itemStr;
}

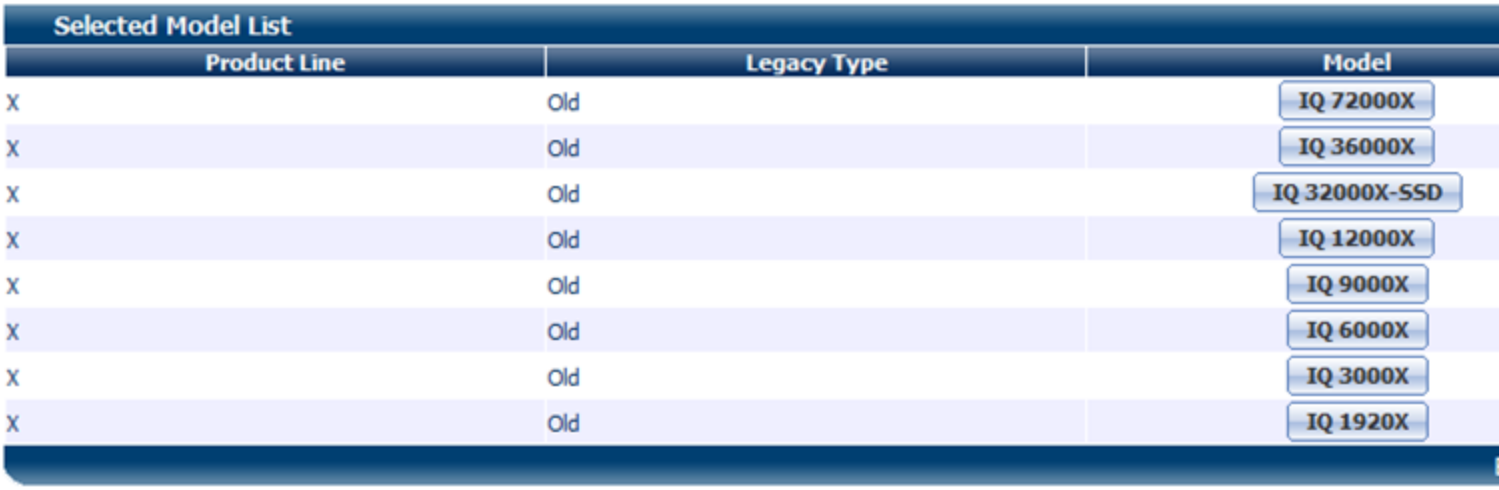
str= str +"</SelectorSearchResults>";

return(str);
```

8. Refresh and deploy the home page. Then, from the home page, enter the Selector. You will see a screen like this:



9. And entering the Search Result screen:



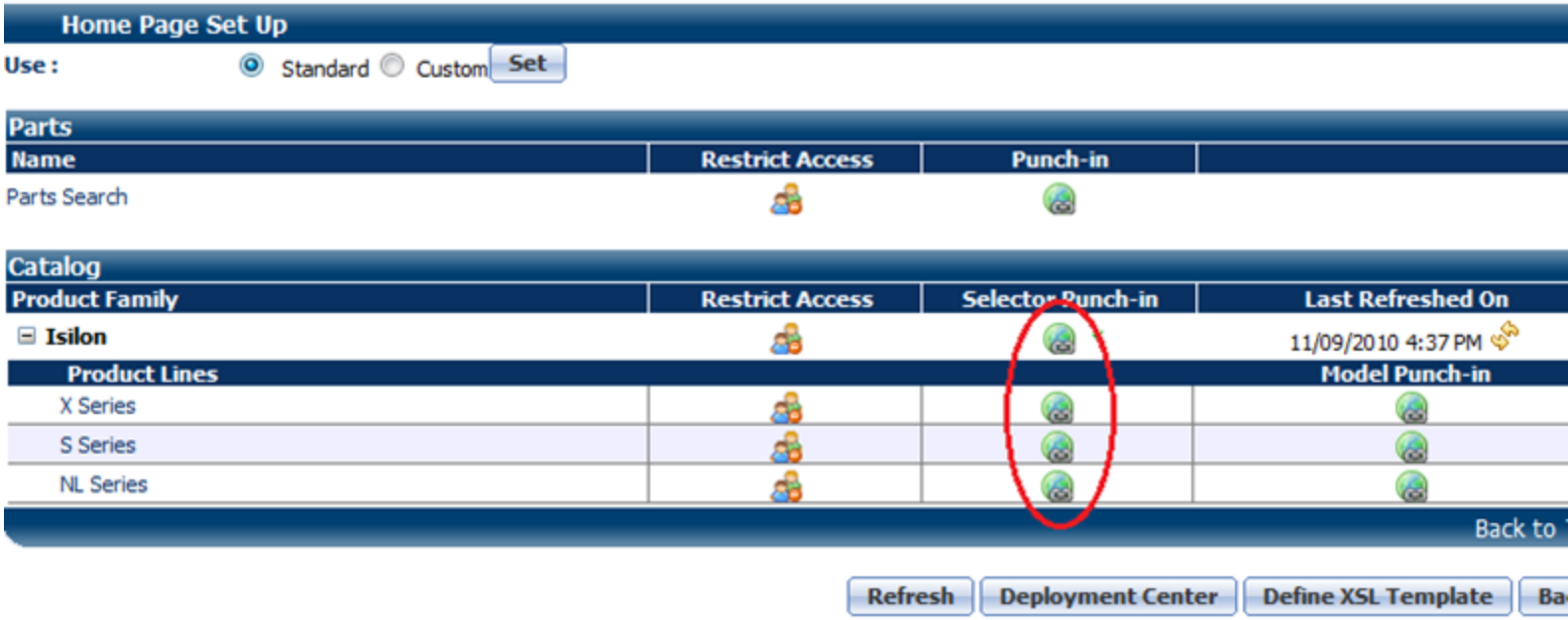


You may need to deploy regularly in configuration if the BMI instance you are working on has its configuration cache turned off.

Step 3: Changing the Homepage UI and the Models that Your Selector Can Access

When you create or modify your search flow, choosing its Type affects the corresponding layout on the Homepage to a certain degree. In most cases, you will probably choose "Choose models across product lines". Upon selecting this option and deploying the Homepage, a selector will be generated for each product line. So if you have 7 product lines, there will be 7 punch-ins to 7 selectors, each of which is only capable of accessing the models that belong to that specific product line. This is probably not what you had in mind. Additionally, you are probably averse to having what looks like 7 identical punch-ins cluttering the Homepage.

By default, a selector punch-in is generated on the Homepage only for each product line. So, you will need to go to Homepage admin and create a selector punch-in for the entire product family.



Note the items circled in red. There is a selector punch-in for each product line in the product family, and by default this is probably what gets generated when you first refresh and deploy the home page. In order to create a selector that can access all models across all product lines, you need to create a selector punch-in for the Product Family. In this example, you will have to click on the selector punch-in icon that corresponds to "Isilon". This will bring you to the selector punch-in list for the Isilon product family:

In this example, a punch-in for the product family has already been created. In your case, you will probably need to create a new one. Name it anything you like.

You may run into an error message that pops up saying something similar to "search group not available for this selector". A work-around for this is to go back to the Search Flow and change its type to "Select product lines". You will have to unselect all product lines. Then, deploy config. Go back to Homepage admin, create a selector the entire product family (this should now work). Once that is done and the homepage is deployed, go back to the search flow and change its Type back to what it was before (don't forget to select the product lines again).

RELATED TOPICS

CONFIGURATION SETTINGS

OVERVIEW

Use this page to customize the configuration page. There are configuration page look and feel options such as the ability to control the display of the processing dialog as well as the number of tabs.

Configurable price options allow the ability to control the display of price on all pages, hide the price when it is zero, as well as display the sum of recommended items, if any. Recommended item options include the ability to control the display of items on the model configuration page, hide missing items, and selection of items by default.

Configuration Cache

Configuration cache is highly recommended (but optional) feature of BigMachines. Support for configuration cache can be enabled by the System fullaccessuser.

The purpose of configuration cache is to store configuration information in memory to improve performance. Since trips to the database server are avoided using the cache, this is a highly optimal solution for most implementations. When configuration cache functionality is enabled, all areas of the system that call configuration methods should be quicker to access after the cache is built.

When configuration cache is enabled, then all changes to catalog data must be deployed to ensure that the cache is synchronized with configuration data. If changes are made to rules, products or flows within a product family and the product family is not deployed, then this will result in a user working off of a cache that is out of date with the actual catalog data.

Steps

1. On the Supported Product Families page, choose the product family you want to deploy.
2. Navigate to the Deployment Center by clicking on List.



By default, the Configuration Cache is disabled for all new product families. Once enabled, Config changes will only be reflected once deployed.

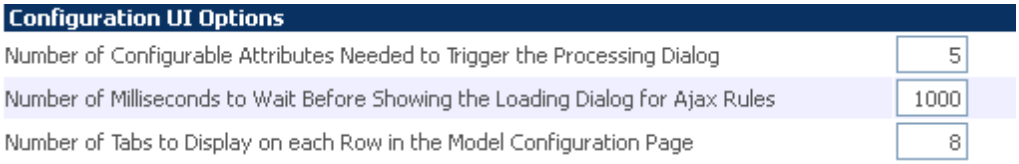
ADMINISTRATION

Setting Configuration Options

You can customize the look and feel of your Configuration page.

1. Navigate from: Admin Home Page > Products > Configuration Settings
2. Enter the number of configurable attributes needed to trigger the processing dialog.

3. Enter the number of milliseconds to wait before showing the loading dialog for AJAX rules:
    i. **A positive number:** This will determine how many milliseconds the dialog will appear.
    ii. **A negative number:** This will disable the processing dialog for AJAX rules.
    iii. **Zero (0):** The dialog will display instantaneously.
4. Enter the number of tabs to display on each row in the model configuration page.



5. Set Configurable Price Options:
    i. **Display price on all pages**: Select Yes to display price on all pages, and No to reject.
    ii. **Hide zero total price**: Select Yes to hide if the total price is zero, and No to reject.
    iii. **Display total sum of recommended items**: Select Yes to display the total sum of recommended items, and No to reject. If you want to display the total sum of recommended items along with a subtotal of mandatory items, select Yes with Subtotal of Mandatory Items.



6. Set Recommended Item Options:
    i. **Display items on end page**: Select Yes if you want to display items on end page, and No if you do not want. To hide the item, select Hide, to hide only the main model, select Hide Main Model.
    ii. **Hide missing spares:** Select Yes if you want to hide missing spares, and No if you do not want.
    iii. **Select recommended items by default**: Select Yes if you want to select recommended items by default, and No if you do not want.
    iv. **Sum Recommended Item Quantities**: Select Yes if you want to sum the recommended item quantities and No if you do not want them to be summed. If this is set to No, the quantities for the same item, that is recommended by different rules, will not be summed and these items will appear as separate line items in the commerce document. Quantities are still summed if the same item is recommended multiple times by the same rule or by different rules with the same variable name.



7. Set Configuration Save Options:

    i. **Enable auto save of Pending Configurations**: Select Yes if you'd like enable this functionality and No to leave it disabled. Note: This is disabled by default.



8. Please see Pending Configuration article for more information.

9. Set Configuration Deploy Options:

    i. **Recreate buyside cache after deployment**: Select Yes if you'd like the buy-side cache to be purged when configuration is deployed. This will ensure in-flight configurations will have their cache rebuilt and the server will then release the original memory that was allocated for the first cache. Note: This is disabled by default.



10. Click Apply to save your changes and remain on the current page. Click Update to save your changes and go back to the Admin home page. Click Back to go back to the Admin home page.

RELATED TOPICS

CONFIGURATION SETTINGS

OVERVIEW

Use this page to customize the configuration page. There are configuration page look and feel options such as the ability to control the display of the processing dialog as well as the number of tabs.

Configurable price options allow the ability to control the display of price on all pages, hide the price when it is zero, as well as display the sum of recommended items, if any. Recommended item options include the ability to control the display of items on the model configuration page, hide missing items, and selection of items by default.

Configuration Cache

Configuration cache is highly recommended (but optional) feature of BigMachines. Support for configuration cache can be enabled by the System fullaccessuser.

The purpose of configuration cache is to store configuration information in memory to improve performance. Since trips to the database server are avoided using the cache, this is a highly optimal solution for most implementations. When configuration cache functionality is enabled, all areas of the system that call configuration methods should be quicker to access after the cache is built.

When configuration cache is enabled, then all changes to catalog data must be deployed to ensure that the cache is synchronized with configuration data. If changes are made to rules, products or flows within a product family and the product family is not deployed, then this will result in a user working off of a cache that is out of date with the actual catalog data.

Steps

1. On the Supported Product Families page, choose the product family you want to deploy.
2. Navigate to the Deployment Center by clicking on List.



By default, the Configuration Cache is disabled for all new product families. Once enabled, Config changes will only be reflected once deployed.

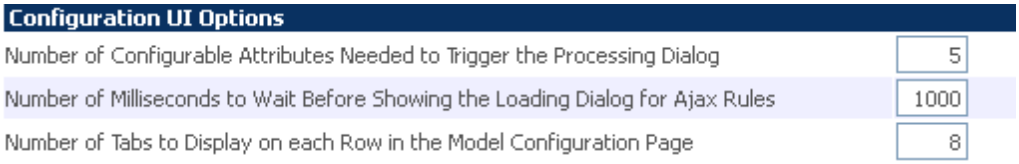
ADMINISTRATION

Setting Configuration Options

You can customize the look and feel of your Configuration page.

1. Navigate from: Admin Home Page > Products > Configuration Settings
2. Enter the number of configurable attributes needed to trigger the processing dialog.

3. Enter the number of milliseconds to wait before showing the loading dialog for AJAX rules:
    i. **A positive number:** This will determine how many milliseconds the dialog will appear.
    ii. **A negative number:** This will disable the processing dialog for AJAX rules.
    iii. **Zero (0):** The dialog will display instantaneously.
4. Enter the number of tabs to display on each row in the model configuration page.



5. Set Configurable Price Options:
    i. **Display price on all pages**: Select Yes to display price on all pages, and No to reject.
    ii. **Hide zero total price**: Select Yes to hide if the total price is zero, and No to reject.
    iii. **Display total sum of recommended items**: Select Yes to display the total sum of recommended items, and No to reject. If you want to display the total sum of recommended items along with a subtotal of mandatory items, select Yes with Subtotal of Mandatory Items.



6. Set Recommended Item Options:
    i. **Display items on end page**: Select Yes if you want to display items on end page, and No if you do not want. To hide the item, select Hide, to hide only the main model, select Hide Main Model.
    ii. **Hide missing spares:** Select Yes if you want to hide missing spares, and No if you do not want.
    iii. **Select recommended items by default**: Select Yes if you want to select recommended items by default, and No if you do not want.
    iv. **Sum Recommended Item Quantities**: Select Yes if you want to sum the recommended item quantities and No if you do not want them to be summed. If this is set to No, the quantities for the same item, that is recommended by different rules, will not be summed and these items will appear as separate line items in the commerce document. Quantities are still summed if the same item is recommended multiple times by the same rule or by different rules with the same variable name.



7. Set Configuration Save Options:

    i. **Enable auto save of Pending Configurations**: Select Yes if you'd like enable this functionality and No to leave it disabled. Note: This is disabled by default.



8. Please see Pending Configuration article for more information.

9. Set Configuration Deploy Options:

    i. **Recreate buyside cache after deployment**: Select Yes if you'd like the buy-side cache to be purged when configuration is deployed. This will ensure in-flight configurations will have their cache rebuilt and the server will then release the original memory that was allocated for the first cache. Note: This is disabled by default.



10. Click Apply to save your changes and remain on the current page. Click Update to save your changes and go back to the Admin home page. Click Back to go back to the Admin home page.

RELATED TOPICS

PENDING CONFIGURATIONS

OVERVIEW

You have the option to auto-save pending configuration. This functionality is designed to save incomplete configurations. This link will appear in the sub-header call Pending Configurations and will take the user to a Configurations list page where she can select a configuration to continue. Incomplete configurations can happen if you are accidentally logged out of your sessions, if you lose a connection or if you're timed out.



What does incomplete mean?

You will start in the exact same configuration flow, in the exact same state.

When are the configurations saved?

Configuration drafts are saved every time an attribute is updated.

How are the configurations saved?

There is one draft, per user, per model that is saved. If you configure two different models, you will see two models appear as a pending configuration. These drafts are not automatically cleared by the application. Users will either need to move forward the configuration or they can delete the drafts.



If a user logs back in and begins to configure a model from the home page, before they click the Pending Configurations link, the new configuration will become the new 'draft' and you will lose the previous draft. Also, if a user proceeds to commerce, or invokes a commerce action (Add to Cart, Add to Favorites, Add to Quote, etc), they will no longer have a draft of the configuration.

ADMINISTRATION

Managing Pending Configurations

1. Navigate from: Products > Configurations Settings.
2. **Enable auto save of Pending Configurations:** Click **Yes** if you'd like enable this functionality and **No** to leave it disabled.



This is disabled by default.

NOTES

Pending configurations will work with active constraints.

An AJAX call saves as the user makes any change to an attribute. When a disconnection occurs, the configuration will have been saved up to the most recent state.

BigMachines doesn't support drafts for uncached segments.

Drafts will last until a user configuration the same model, the pending configuration is added to a quote or the product is deployed.

If you have multiple configuration flows running and the end user clicks on PREVIOUS, they will lose their configuration flow and will have to re-populate.

THIS IS NOT TO BE USED AS FAVORITES!

RELATED TOPICS

SOAP CONFIGURATION

OVERVIEW

You can generate a skeleton input for each operation in the configuration WSDL, which contains five operations: configure, getBOM, getPrice, getSpare and deploy. For each operation, there are three fields that are always required in the header: sessionId, category: Configuration, and schemaLocation.

View API Details

API	Description	Required Input Variable	Success Response Variable	Fail Response Variable

configure	Takes input of a particular product family, line and model and generates a configuration. The response changes based on the options specified. For example, if SOAP XML mandates generating a transaction for the given configuration, then a transaction of the specified process is included in the response. Similarly, other details can be chosen to be included or not included in the response.	• sessionId
		• category
		• schemaLocation
		• segment
		• product_line
		• model
		• var_name attribute
		• price_book
		• var_name
			• status
			  ○ success
			  ○ message
			• item
			  ○ segment
			  ○ product_line
			  ○ model
				excetionCode
				exceptionMessage

getBom	Returns the BOM for the model.	• sessionId
		• category
		• schemaLocation
		• segment
		• product_line
		• model
			• status
			  ○ success
			  ○ message
			• item
			  ○ segment
			  ○ product_line
			  ○ model
			• attributes
			• flow
			• bom
			  ○ rule
			  ■ item
			  ■ *partNumber*
			  ■ *quantity*
				excetionCode
				exceptionMessage

getPrice	Return the base price of the model.	• sessionId
		• category
		• schemaLocatio
		• segment
		• product_line
		• model
			• status
			  ○ success
			  ○ message
			• item
			  ○ segment
			  ○ product_line
				excetionCode
				exceptionMessage

API	Description	Required Input Variable	Success Response Variable	Fail Response Variable
			○ model • attributes • flow • price ○ totalPrice ○ attribute ■ value	
getS-pare	Returns the recommended items for the model.	• sessionId • category • schem-aLocation • segment • product_line • model • price_book_var_name	• status ○ success ○ message • item ○ segment ○ product_line ○ model • attributes • flow • spare ○ rule ■ comments ■ item ■ *man-datory* ■ *quant-ity*	excetionCode excep-tionMessage
deploy	Deploys a specific product family.	• sessionId • category • schem-aLocation • product_fam-ily • var_name • attribute	• status ○ success ○ message	excetionCode excep-tionMessage

Returning Specific Attributes in Configuration APIs

You can return specific attributes from a web service request with the `<bm:return_specific_attributes />` tag. **Example:** you can pull in four main-document attributes and two sub-document attributes and a shortened version of the transaction XML, displaying only those attributes, will be returned.



This will only be allowed with the configure API.

ADMINISTRATION

Accessing Users WSDL & generating skeleton input

1. Navigate from: Admin Home Page > Integration Platform > Web Services.
2. Select the **Configuration** tab.
3. Choose a configuration API from the **Select API** drop-down menu.
4. Select **one** of the following:
    - Punchin URL and enter the URL, or
    - Product Family and choose the product family, product line, and model from the corresponding drop-down menus.
5. Click **Generate Input** to retrieve the skeleton.

Examples of Configuration APIs

configure - Input SOAP XML

The following is the Minimum XML Input needed to configure a model only (will not create a commerce transaction or a quote, this XML also excludes the Price Book Tag):

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:configure xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
</bm:item>
<bm:responseIncludes>
<bm:transaction>
```

```
<bm:process_var_name>standard_commerce_process</bm:process_var_name> <!--This refers to the variable name of the commerce process.-->

<bm:document_var_name>quote_line</bm:document_var_name> <!--This refers to the variable name of the commerce document. It's commonly referred to as either Quote (main) or Line (sub)-->

<bm:id>21185424</bm:id> <!--Transaction Number -->

</bm:transaction>

</bm:responseIncludes>

<bm:price_book_var_name>_default_price_book</bm:price_book_var_name> <!--This is included if you need to create a transaction or a quote if the price book is enabled. -->

<bm:attributes/>

</bm:configure>

</soapenv:Body>

</soapenv:Envelope>
```

getBOM - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Header> <bm:userInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>

</bm:userInfo>

<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>

<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/config/hiTech/managedHosting_managedHosting.xsd</bm:schemaLocation>

</bm:xsdInfo>

</soapenv:Header>

<soapenv:Body>

<bm:getBom xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<bm:item>

<bm:segment>hiTech</bm:segment>

<bm:product_line>managedHosting</bm:product_line>

<bm:model>managedHosting</bm:model>

</bm:item>

<bm:attributes/>

</bm:getBom>

</soapenv:Body>

</soapenv:Envelope>
```

getPrice - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Header>
```

```
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getPrice xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
</bm:item>
<bm:attributes/>
</bm:getPrice>
</soapenv:Body>
</soapenv:Envelope>
```

getBom - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getBom xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
</bm:item>
```

```
<bm:attributes/>
</bm:getBom>
</soapenv:Body>
</soapenv:Envelope>
```

getSpare - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getSpare xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
<bm:mandatory />
<bm:quantity />
</bm:item>
<bm:price_book_var_name>_default_price_book</bm:price_book_var_name>
<bm:attributes/>
</bm:getSpare>
</soapenv:Body>
</soapenv:Envelope>
deploy - Input SOAP XML
Note: Deploy will is available for only
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
```

```
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:deploy xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:product_family>survey</bm:product_family>
<bm:emailTo/>
</bm:deploy>
</soapenv:Body>
</soapenv:Envelope>
```

deploy - Input SOAP XML

When you select deploy, you'll no longer have the option to select Punchin URL, you will just select the Product Family you wish to deploy.

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_
0/config/hiTech/managedHosting_managedHosting.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getBom xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item> <bm:segment>hiTech</bm:segment>
<bm:product_line>managedHosting</bm:product_line>
<bm:model>managedHosting</bm:model>
</bm:item>
<bm:attributes/>
</bm:getBom>
</soapenv:Body>
</soapenv:Envelope>
```

RELATED TOPICS

PRICE BOOKS OVERVIEW

OVERVIEW

Parts are defined along with a price. Price Books offer an additional way to associate parts with price. One party can be associated with multiple price books. The fullaccessuser can specify the price book applicable depending on the user type.

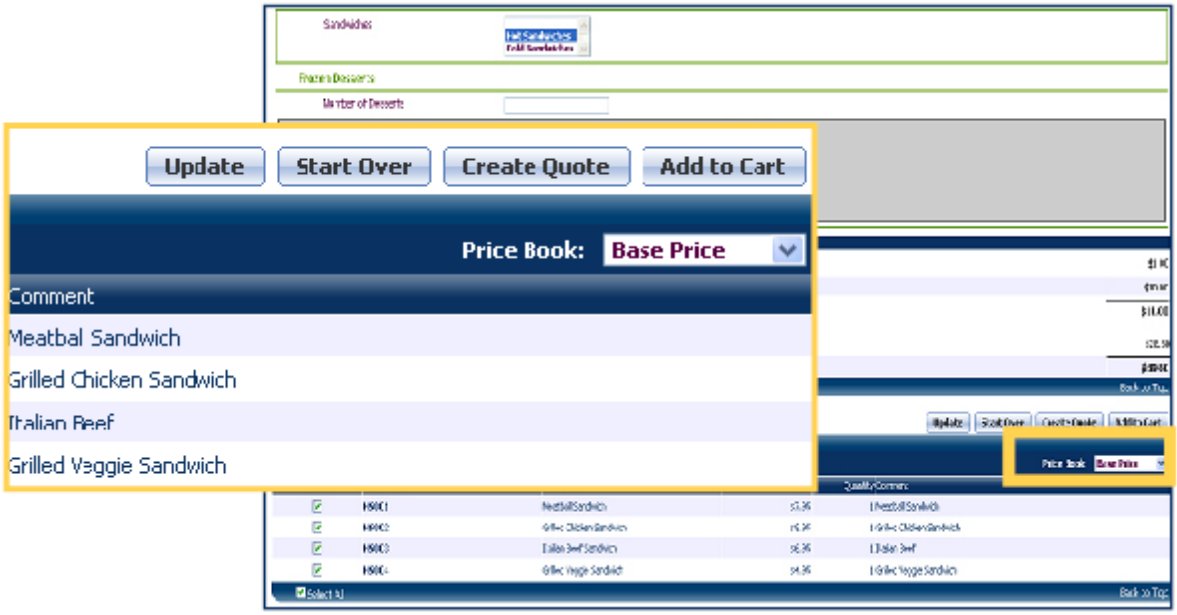
Example: Consider that there are two prices books: US Price Book and Germany Price Book. If a user based in New York decides to purchase a certain part, then the "US Price Book" price book will be used to determine the price of the part. Similarly, if a German user wants to purchase the same party, then the "German Price Book" will be used to set the price of the part. Therefore, the same part number can have different prices based on the price book that is applicable for a particular user. Please see Creating and Maintaining Part Filters for more information on how to set-up a similar use case.

ADMINISTRATION

User-Side Price Books

Configuration

In Configuration, a Price Book drop-down menu will appear if you have more than one price book available.



Parts Search

The User will only be able to select a Price Book if they have been given access to it. This can be achieved by creating filters.



Parts Export & Price Book Associations

Parts Export no longer includes Price Book Associations columns.

- If users wish to export Price Book Associations, they may do so separately.
- Users may still import older Excel documents that contain Price Book Associations columns.

NOTES

You can define rules with price books as a rule input. **Example:** you can use a recommendation rule to force set a price book or constrain a price book. Constrained price books are removed from the price book list unless they are currently selected, in which case they will appear marked. You could also use a price book as a rule condition for a hiding rule and hide particular attribute(s) when a certain price book is selected.

If a part is not associated with a price book, then the price of the part is determined by the price specified when creating the part.

When five or more currencies are being used, fields will be paginated into columns to eliminate horizontal scrolling. Column headers will repeat at the bottom of the table for reference in long data sets.

Standard Price Books can accommodate as many currencies as are supported on a BigMachines site.

Price Books are not enabled for all applications. If you want to enable a Price Book for your application, please contact your System fullaccessuser.

RELATED TOPICS

PRICE BOOKS OVERVIEW

OVERVIEW

Parts are defined along with a price. Price Books offer an additional way to associate parts with price. One party can be associated with multiple price books. The fullaccessuser can specify the price book applicable depending on the user type.

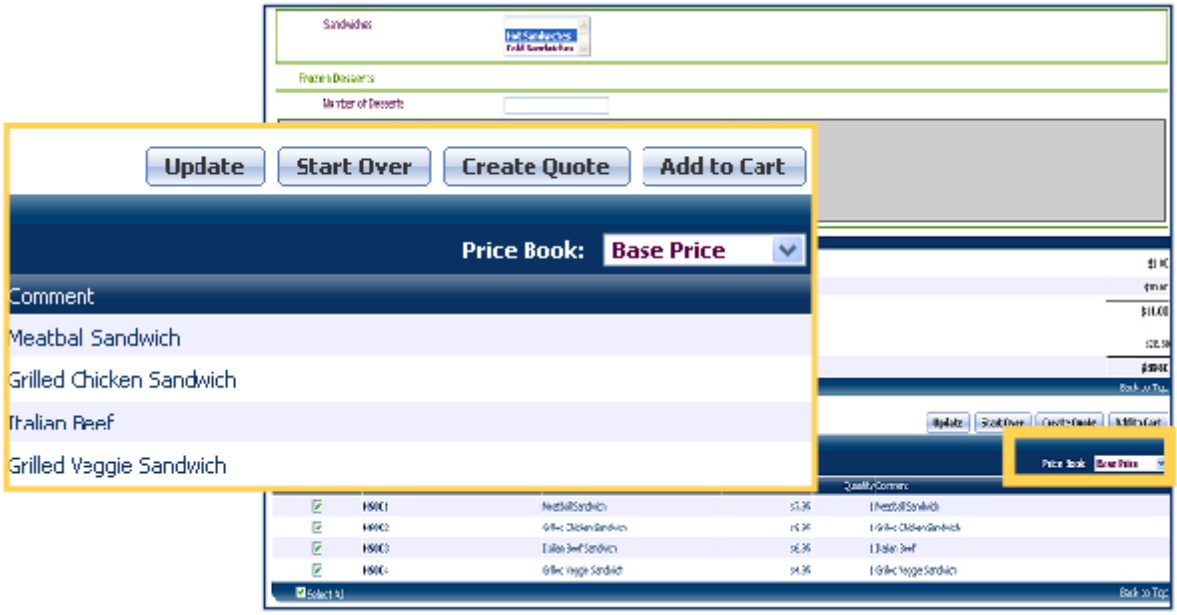
Example: Consider that there are two prices books: US Price Book and Germany Price Book. If a user based in New York decides to purchase a certain part, then the "US Price Book" price book will be used to determine the price of the part. Similarly, if a German user wants to purchase the same party, then the "German Price Book" will be used to set the price of the part. Therefore, the same part number can have different prices based on the price book that is applicable for a particular user. Please see Creating and Maintaining Part Filters for more information on how to set-up a similar use case.

ADMINISTRATION

User-Side Price Books

Configuration

In Configuration, a Price Book drop-down menu will appear if you have more than one price book available.



Parts Search

The User will only be able to select a Price Book if they have been given access to it. This can be achieved by creating filters.



Parts Export & Price Book Associations

Parts Export no longer includes Price Book Associations columns.

- If users wish to export Price Book Associations, they may do so separately.
- Users may still import older Excel documents that contain Price Book Associations columns.

NOTES

You can define rules with price books as a rule input. **Example:** you can use a recommendation rule to force set a price book or constrain a price book. Constrained price books are removed from the price book list unless they are currently selected, in which case they will appear marked. You could also use a price book as a rule condition for a hiding rule and hide particular attribute(s) when a certain price book is selected.

If a part is not associated with a price book, then the price of the part is determined by the price specified when creating the part.

When five or more currencies are being used, fields will be paginated into columns to eliminate horizontal scrolling. Column headers will repeat at the bottom of the table for reference in long data sets.

Standard Price Books can accommodate as many currencies as are supported on a BigMachines site.

Price Books are not enabled for all applications. If you want to enable a Price Book for your application, please contact your System fullaccessuser.

RELATED TOPICS

PRICING RULES

OVERVIEW

Pricing rules are used to calculate price based on how a product is configured. Pricing rules can be based on a combination of one or more configured values. Like most rules, a pricing rule also has a condition and an action. The action determines the price of the model being configured.

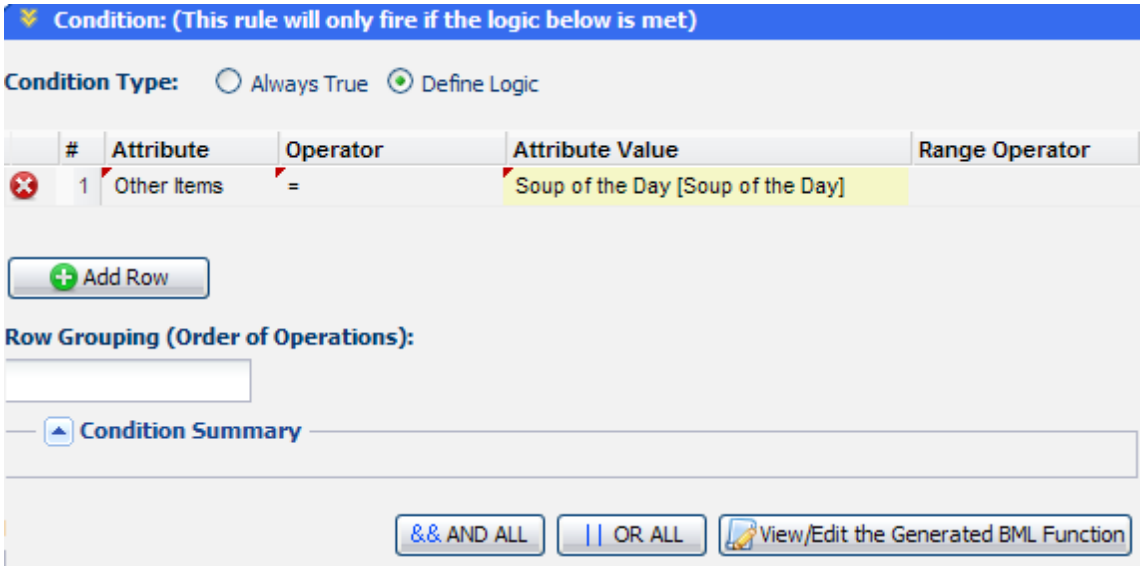
Displaying the price to the user is the default setting, though doing so is optional. If the price should be shown to the user can be determined by an Advanced Function. If the function returns true then the price is displayed. If the function returns false, then the price is not displayed. Including the price in the total configured price is also optional. The default value is true, which indicates to the system that the price should be included in the total configured price. An advanced function can modify a default value if necessary.

ADMINISTRATION

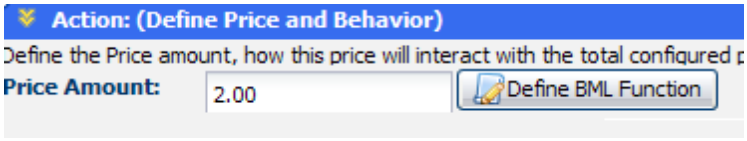
Adding a Pricing Rule

Scenario: You own a pizza company and you need to set the price for the Soup of the Day. According to the menu, the price of the soup is $2.00.

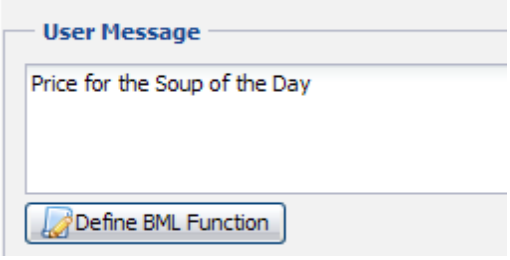
1. Navigate from: Product Family > List of Pricing Rules.
2. Click **Add** to create a new rule.
3. Enter the basic properties of the rule.
4. Create the Conditions. *This rule should run ONLY when a user selects Soup of the Day.*



5. Define the **Price Amount** of the Action. *This is how the price will interact with the total configured price.*



6. Enter a **User Message**. *This will be displayed with the price.*



7. Define the **Behavior** of the Action. *Decide on the following options:*
   - Do you want the price of the item to be included in the total price?
   - Do you want to display the price of this item to the user?
   - Do you want to hide zero values from the user? If no, then a value of $0 will be displayed.



When the user decides to place an order and they select Soup of the Day, the price will be included with the total configured price of the model.

Price	
Price for a Small Thin Crust Pizza	$10.00
Price for the Soup of the Day	$2.00
Total Configured Price of Model	**$12.00**

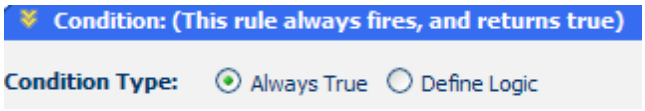
Editing a Pricing Rule

You can edit the Main Information by making changes to the **Name, Description, Status and Start/End Dates**. If the rule is defined at the Product Family level, you can choose the Area of the application. The areas available for application are Configuration, Selection or Both. If Configuration is selected, the rule is only applicable during configuration. If Selection or Both are chosen, then the rule is applicable in the search flow as well.

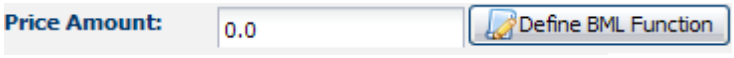
Scenario: You will be setting the prices for the pizzas based on their size and crust type.You will also need to create a user message that explains to the user exactly what they are ordering. According to the menu, you have the following options and prices:

- Small Thin Crust - $9.00
- Medium Thin Crust - $12.00
- Large Thin Crust - $15.00
- Medium Deep Dish - $17.00
- Large Deep Dish - $20.00

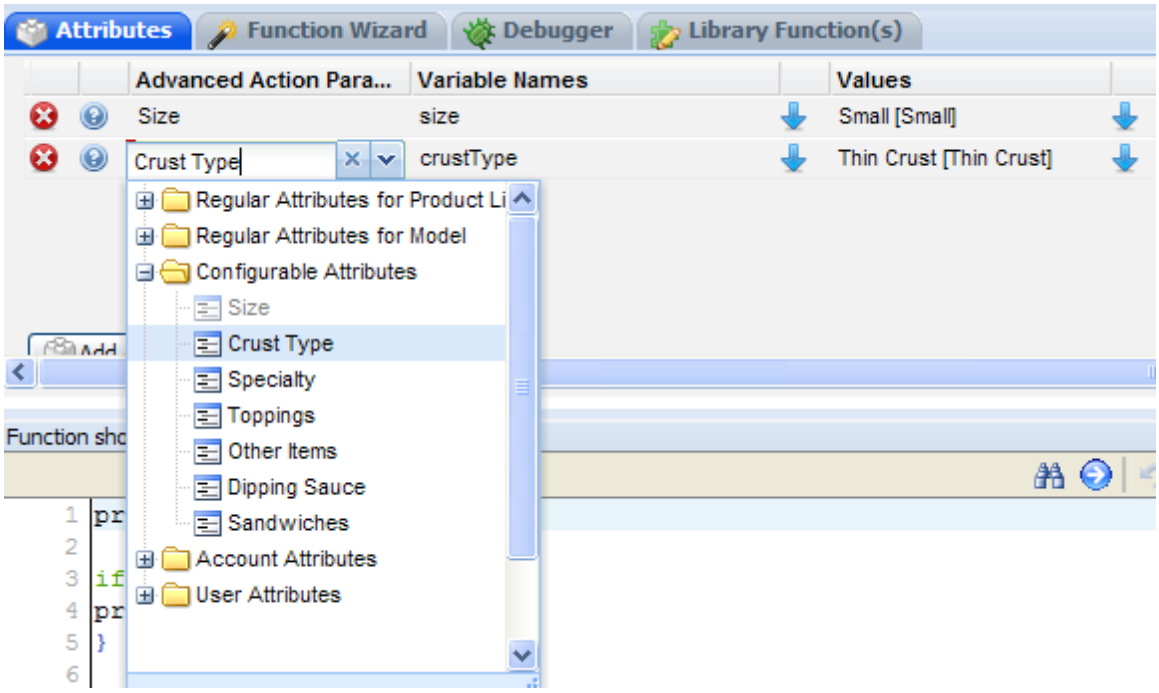
1. Navigate from: Product Family > List of Pricing Rules.
2. Click **Add** to create a new rule.
3. Enter the basic Properties of the rule.
4. Create your **Condition Type**. This rule can run all of the time, so it should be set to **Always True**.



5. Define the **Price Amount** of the Action. Since you are creating a rule for multiple items, you will need to create an Advanced Function by clicking **Define BML Function**.



6. Click **Add Attributes**.
7. Choose **Size** and **Crust Type**from the drop-down menu, since the pricing rule is based on these attributes.



8. Begin writing the pricing script in BML. *You will need to first assign a variable to zero. Try* `price = 0;`

```
price=0;

//Setting price for small pizza
if(size=="Small"){
        price=10;

                //Setting price of Medium Thin Crust Pizza
                }elif(size=="Medium" AND crustType=="Thin Crust"){
                        price=12;

                //Setting price of Medium Deep Dish Pizza
                }elif(size=="Medium" AND crustType=="Deep Dish"){
                        price=17;

                //Setting price of Large Thin Crust Pizza
                }elif(size=="Large" AND crustType=="Thin Crust"){
                        price=15;

                //Setting Price of Large Deep Dish Pizza
                }elif(size=="Large" AND crustType=="Deep Dish"){
                        price=20;
}

return price;
```

9. Click **Define BML** to create the user message the buyer will see.

10. Select the necessary attributes. They should be the same as those found in **Step 7**.

11. Create an empty string variable to hold your user message string.

12. Write the user message to be returned.

```
message = "";

message = "Price for a ";
message = message + size + " " + crustType + " " + "Pizza";

return message;
```



In the **getPrice() SOAP** call, the Hide Zero Price behavior is replicated by using the attribute **bm:hideZeroPrice**.

NOTES



During reconfiguration, pricing rules are always evaluated, regardless of whether the condition attributes are in the flows.

RELATED TOPICS

PRICE BOOKS OVERVIEW

OVERVIEW

Parts are defined along with a price. Price Books offer an additional way to associate parts with price. One party can be associated with multiple price books. The fullaccessuser can specify the price book applicable depending on the user type.

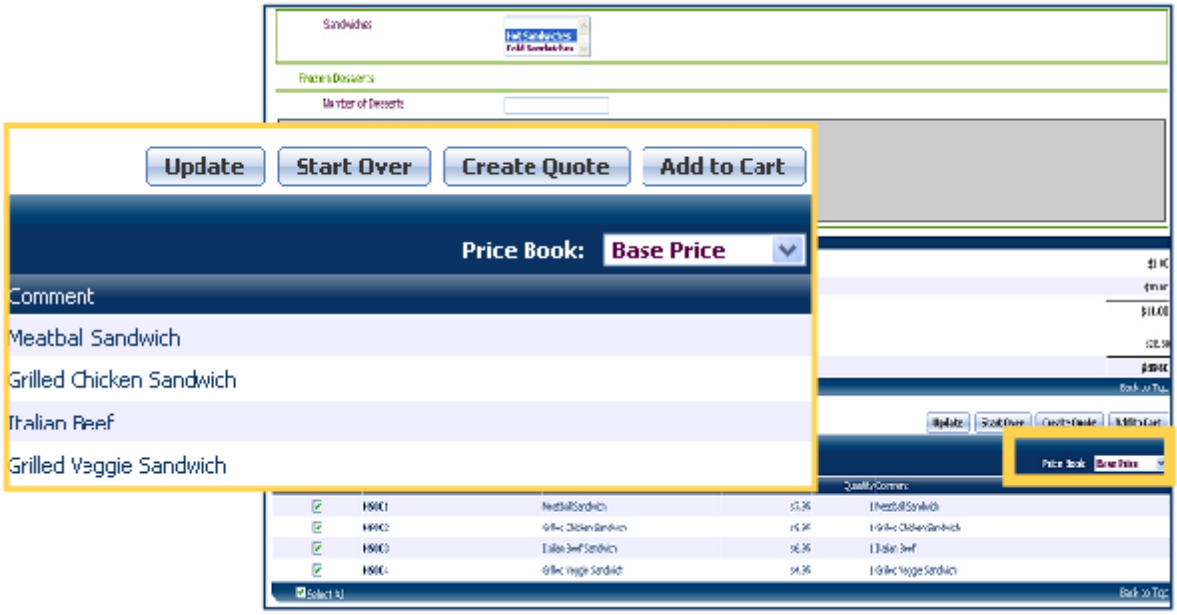
Example: Consider that there are two prices books: US Price Book and Germany Price Book. If a user based in New York decides to purchase a certain part, then the "US Price Book" price book will be used to determine the price of the part. Similarly, if a German user wants to purchase the same party, then the "German Price Book" will be used to set the price of the part. Therefore, the same part number can have different prices based on the price book that is applicable for a particular user. Please see Creating and Maintaining Part Filters for more information on how to set-up a similar use case.

ADMINISTRATION

User-Side Price Books

Configuration

In Configuration, a Price Book drop-down menu will appear if you have more than one price book available.



Parts Search

The User will only be able to select a Price Book if they have been given access to it. This can be achieved by creating filters.



Parts Export & Price Book Associations

Parts Export no longer includes Price Book Associations columns.

- If users wish to export Price Book Associations, they may do so separately.
- Users may still import older Excel documents that contain Price Book Associations columns.

NOTES

You can define rules with price books as a rule input. **Example:** you can use a recommendation rule to force set a price book or constrain a price book. Constrained price books are removed from the price book list unless they are currently selected, in which case they will appear marked. You could also use a price book as a rule condition for a hiding rule and hide particular attribute(s) when a certain price book is selected.

If a part is not associated with a price book, then the price of the part is determined by the price specified when creating the part.

When five or more currencies are being used, fields will be paginated into columns to eliminate horizontal scrolling. Column headers will repeat at the bottom of the table for reference in long data sets.

Standard Price Books can accommodate as many currencies as are supported on a BigMachines site.

Price Books are not enabled for all applications. If you want to enable a Price Book for your application, please contact your System fullaccessuser.

RELATED TOPICS

MANAGING PRICE BOOKS

OVERVIEW

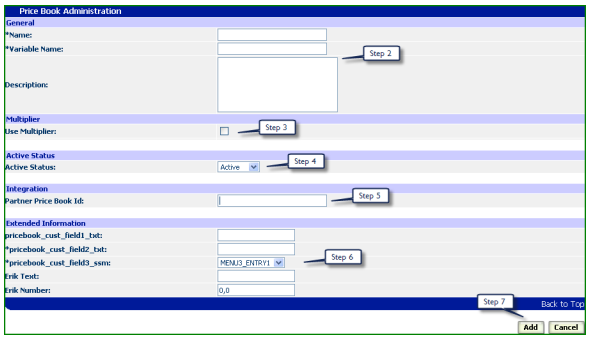
When price books are enabled for the site, then a Default Price Book is automatically made available (it can be viewed in the Price Book Administration List page). The default price book contains the price associations of a part with the price that is defined during part creation.

ADMINISTRATION

Managing Price Books

Adding a Custom Price Book

1. Follow the Standard Navigation Path and click **Add** to proceed to the Price Book Administration page.
2. Enter a **Name**, **Variable Name**, and **Description** for your Price Book.
3. Select the **Use Multiplier** checkbox, if necessary. *The multiplier will automatically populate the custom part book with the price associations between parts and the specified multiple of the base price.*



4. Choose **Active** or **Inactive** from the Active Status drop-down menu.
    - **Inactive Price Book:** cannot be used to determine price for any user.
    - **Active Price book:** sets the price for the appropriate user.
5. Enter the **Partner Price Book Id**, if applicable. For more information, see Partner Parts Integration.
6. Enter **Extended Information** values for the custom price book fields, if necessary.
7. Click **Add**.

Editing a Custom Price Book

1. Select the Price Book you'd like to edit on the Price Book Administration List page.
2. Edit any information that you'd like to modify.

3. Click Update.



Edits take effect after re-deployment.

Ordering a Price Book

1. Enter the desired order number in the **Order** column on the Price Book Administration List page.
2. Click **Update**.

Deleting a Price Book

1. Select the Price Book that you want to delete on the Price Book Administration List page.
2. Click **Delete**.



The Default Price Book cannot be deleted.

Managing Price Book Fields

Custom part fields can be used to create price book filters or to supplement the set of base attributes available for price books. The system supports up to five pre-defined customizable price book fields. Four of these are of type string. One field can be used for numerical value (integer or float). Price book filters let you control access rights to price book. Administrating filters involves classifying parts with certain values and then defining permissions to each classification.

Custom Fields Displayed:

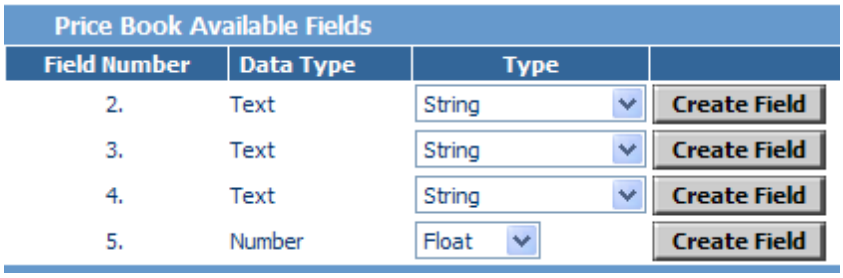
- The Price Book Fields section lists fields that have already been added as custom fields. You can also specify if a field is going to be used as a Filter.
- The Price Book Available Fields section lists the fields that are available for creating custom fields. If any fields have been defined, they will appear in the Defined Fields section.

Adding a Price Book Custom Field

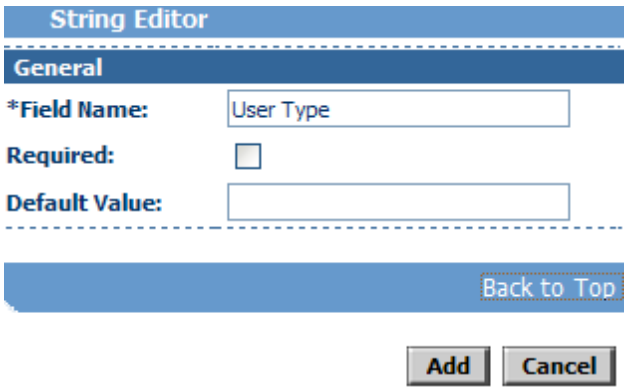
Example: Let's say that you would like to assign access rights to a price book based on user type.

1. Navigate from: Parts > Price Book > Custom Price Book Fields
2. Choose the **Type** using the drop-down menu under Price Book Available Fields.

3. Click **Create Field** to launch the attribute editor page.



4. Add **Field Name** and default value (if required).



5. Click **Add**. *The field will now appear in the Price Book Defined Fields section.*
6. Choose the filter from the Filter/Normal drop-down menu to assign access rights based on user type.
7. Click **Update**.
8. Click **Define Script** to create a script that will return a certain price book based on user type.

Variable	Variable Name	Attribute Type
User Login Name	**_username**	String
Company Login Name	**_company**	String
User Type	**_user_type**	String
Company Type	**_company_type**	String
User Groups	**_user_groups**	String
Customer Id	**_customer_id**	String

BML

Function should return a value of type **String**.

Editor Help

```
if(_company_type == "FullAccessWithESales" and _user_type == "FullAccess"){
return "newPriceBook";
}
return "_default_price_book";
```

Script
Compar
Restrict
FullAcce
BuyAcc
QuickRe



For **Text**, you can choose Text or Single-Select Menu. For **Number**, you can select Float or Integer.



Price Book fields **Add**, **Edit** and **Delete** only take affect after deployment.

Editing a Price Book Custom Field

1. Navigate from: Price Book Defined Fields.
2. Choose the **Type** using the drop-down menu.
3. Set the field either as **Normal** or **Filter**.
4. If the field created is set as **Filter**, write/edit a filter script.
5. Click **Deploy** to deploy the available fields. *Changes must be deployed before they take effect.*



If a field is changed from Filter to Normal, then the script is hidden. If the field type is changed back to Filter, then the script will reappear again in the corresponding Function Editor.



Price Book fields **add**, **edit** and **deletes** only take affect after deployment.

NOTES



Parts Export no longer includes Price Book Associations columns. If users wish to export Price Book Associations, they may do so separately. Users may still import older Excel documents that contain Price Book Associations columns.



A maximum of 30 price books can be defined.

Standard Price Books can accommodate as many currencies as are supported on a BigMachines site.

RELATED TOPICS

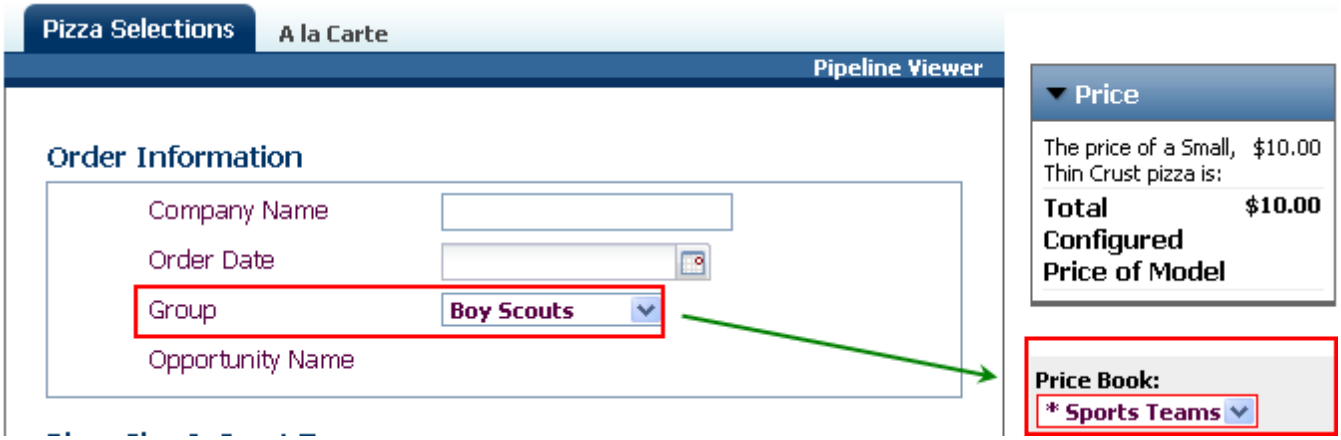
PRICE BOOKS & CONFIGURATION RULES

OVERVIEW

BigMachines allows price books to be used in configuration rules as a variable in rule conditions and actions. **For Example:** you can use a recommendation rule to force set a price book or constrain a price book. Constrained price books are removed from the price book list unless they are currently selected, in which case they will appear marked. You could also use a price book as a rule condition for a hiding rule and hide particular attribute(s) when a certain price book is selected.

Constraint Rule

When a price book is used as the action attribute in a constraint rule, it will be removed from the price book list unless it is currently selected. In the example below, a constraint rule is running that says when Boy Scouts is selected, the Price Book should be set to Scouting. Notice that the current price book is Sports Teams and is marked with an asterisk (*) and is encased in red.

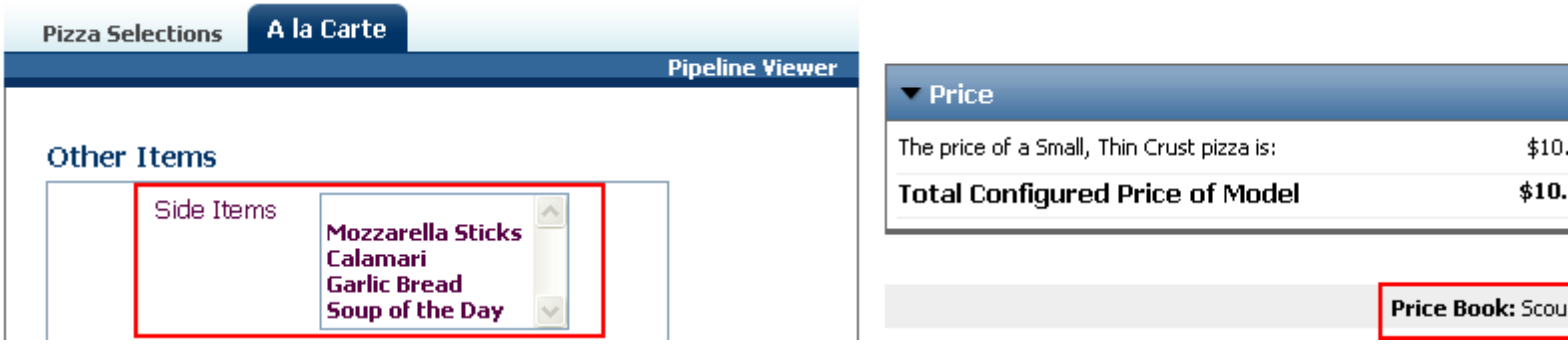


When the user clicks on the Price Book list to select a new price book, you'll notice that the only one available for selection is Scouting.

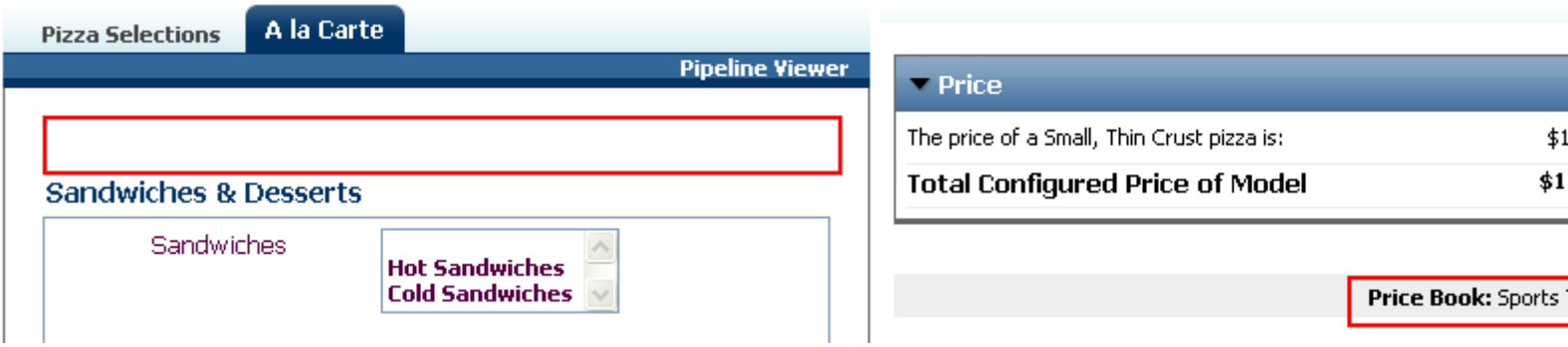


Hiding Rule

When a price book is used as a rule condition in a hiding rule, it can hide attribute(s) based on which price book is selected. In the example below, you'll notice that in the first image, you can see the Side Items attribute displaying.



However, when the price book is changed to Sports Teams, you'll notice that the attribute has been hidden from view.



Recommendation Rule

When a price book is used as the action attribute in a recommendation rule, it will be forced set for the user, meaning they will not be able to change it. In the example below, a recommendation rule is running that says when Girl Scouts is selected as the Group, then the Price Book should be set to Scouting. Notice that it is read only.

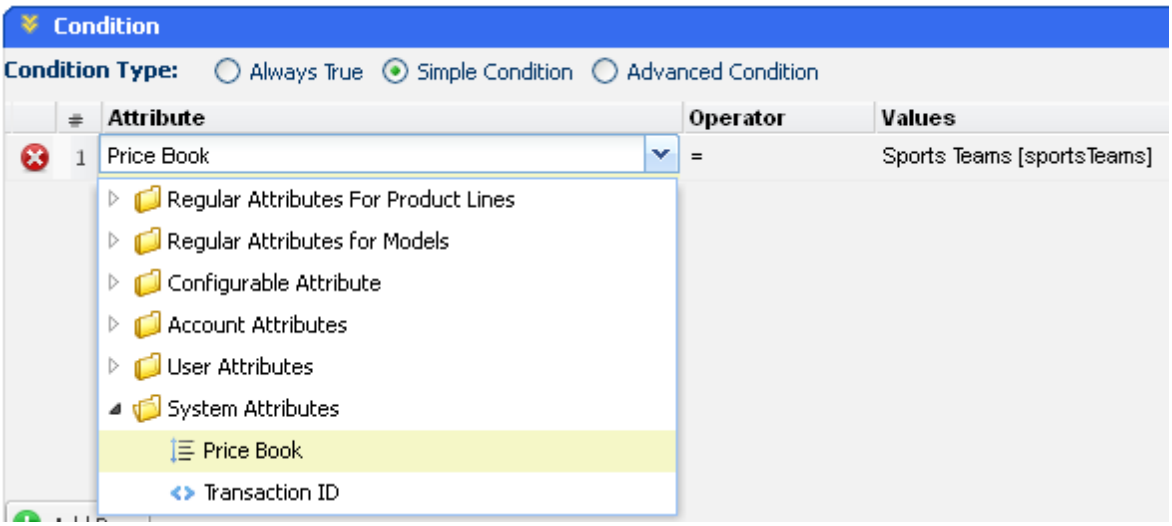


ADMINISTRATION

As previously mentioned, price books are now available for use as rule inputs, whether as a rule condition or as an action attribute. They can also be used in BML scripts. The following steps will explain how to use a select a price book as a rule input.

Selecting a Price Book as a Rule Input

1. Navigate to a configuration rule.
2. Click **Simple Condition**to use a Price Book as a rule condition.
3. Choose **System Attributes**from the attribute drop-down menu.
4. Choose **Price Book** from the options.
5. Select the appropriate value(s) for the Price Book.



NOTES

The price book variable name is `_price_book_var_name`.

Upon Add from Catalog/Reconfigure in Commerce, the transaction price book will take precedences over price books returned from a recommendation rule.

Standard Price Books can accommodate as many currencies as are supported on a BigMachines site.

Price book information is not set when the items is added to Favorites, even if a specific price book has been recommended. On reconfigure, the recommended price book will be set and the user will be allowed to change it.

RELATED TOPICS

CREATING & MAINTAINING A PARTS FILTER

OVERVIEW

The following example script shows how parts filtering can work:

```
/*********************************/
*Logic: Filter parts based on company and user* type.
*Field returned: String
/*********************************/
if(_company_type=="BuyAccess" and _user_type=="FullAccess"){return ""};
return "Acme Parts, Generic Parts";
```

This script example could be used to filter parts based on company type and user type. FullAccess users in the BuyAccess company type see all classifications of parts because all menu values are returned with the double quotes. Any other user type sees parts classified as "Acme Parts" or "Generic Parts".

ADMINISTRATION

Creating a Parts Filter

1. Designate the customized part field as a "filter" field type.
2. Write a script to define the filter. For more information, see Function Wizard.
3. Deploy the changes for the filter to take effect.

Attribute Name	Variable Name	Data Type	Description
User Login Name	_user-name	String	Restrict or allow access to parts by selecting a specific user login to add to your filter script. I.E. if (_username = "joesmith") { return ""};
Company Login Name	_company	String	Restrict or allow access to parts by selecting a company name to add to your filter script. I.E. if (_username = "joesmith" and _company = "BigMachines") { return ""};
User Type	_user_type	String	Restrict or allow access to parts by selecting a user type to add to your filter script: RestrictedAcces, FullAccess, QuickRegistrationBuy, ChannelAgent, SalesAgent and/or BuyAccess. I.E. if (_username = "joesmith" and _company = "BigMachines" and _user_type = "ChannelAgent") { return "" };
Company Type	_company_type	String	Restrict or allow access to parts by selecting a company type to add to your script: RestrictedAccess, FullAccesWithESales, BuyAccess, and/or QuickRegistrationBuy. I.E. if (_username = "joesmith" and _company_type = "FullAccessWithESales") { return ""};
User Groups	_user_groups	String	Restrict or allow access to parts by selecting a user group to add to your filter script. I.E. if (_user_groups = "Girl Scouts) { return Acme Parts; } return "";
Customer ID	_customer_id	String	Restring or allow access to parts by selecting a Customer ID.



You can create a filter by designating a custom field as a "filter" type, and then generating a script that sets filtering rules. Filtering happens when a user enters the system and a filter has been created to show that person a specific set of parts.

Editing a Parts Filter

1. Select the field within Part Defined Fields that you want to use as a filter. *If the field is of a normal type, change the designation to filter, and click* ***Update****.*
2. Click **Edit Script** under Advanced Scripts corresponding to the filter field.
3. Write a filter script.
4. Click **Deploy** to activate the new filter.



The return for a parts filter is a list of allowed values, separated by commas.

Removing a Parts Filter

1. Choose the field whose filter you want to remove, and do either of the following:
   - To shut off the filter: Select **Normal** under the **Filter/Normal** column, and click **Update**.
   - To delete the filter: Select the **Delete** checkbox, and click **Update**.
2. To activate filter changes immediately, click **Deploy**.



You can remove a filter from a defined field in one of two ways. To delete the filter, you can remove it from the system. To shut off the filter, without deleting it, you can toggle the field type from filter to normal.

Use Case: Marking & Removing Inactive Parts

You need to mark some parts as inactive and remove them from use without removing them from the parts database.

1. Add a new single-select menu field using the steps under "Creating a Parts Filter".
2. Create a require field with the options: Active and Inactive. Set the appropriate default value for your case.
3. Write return "active"; in the filter script. The new field will be blank for all existing parts. You must set it to the desired value for ALL EXISTING PARTS.

RELATED TOPICS

MANAGING PRICE BOOK ASSOCIATIONS

OVERVIEW

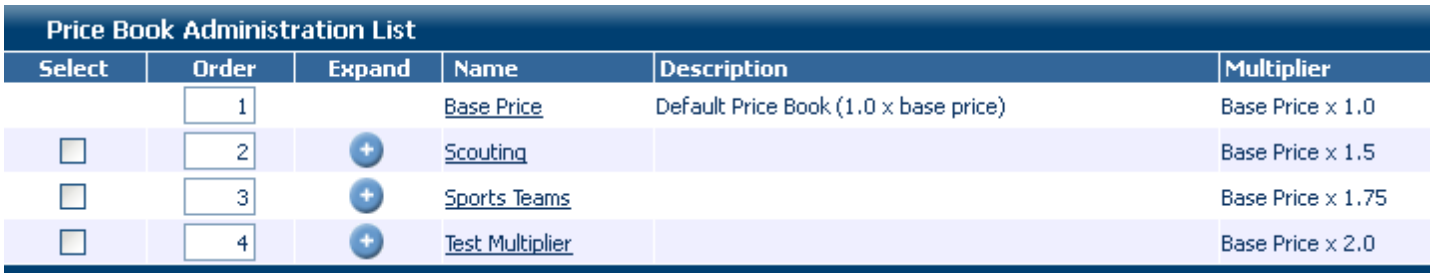
Price book associations allow you to add parts to a custom price book and set their price, if its different from the base price.

Navigation: Admin Home Page > Parts > Price Book > List

ADMINISTRATION

Adding Price Book Associations

1. Click the **Expand** button next to the Price book name.



Price Book Administration List

Select	Order	Expand	Name	Description	Multiplier
	1		Base Price	Default Price Book (1.0 x base price)	Base Price x 1.0
☐	2	⊕	Scouting		Base Price x 1.5
☐	3	⊕	Sports Teams		Base Price x 1.75
☐	4	⊕	Test Multiplier		Base Price x 2.0

2. Click **Add** on the Price Book Associations page.
3. Select the desired Part and click **Associate** on the List of Parts page.
4. Set the associate base price for different supported currencies on the Price Book Association page.
5. Click **Update**.



A part can be associated with several different price books.

Editing Price Book Associations

1. Click the **Expand** button next to the Price book name.
2. Search for the part number and edit the price on the Price Book Associations page.
3. Click **Update**.

Deleting Price Book Associations

1. Click the **Expand** button next to the Price book name.
2. Select the checkbox of the part number you would like to delete.

3. Click **Delete**.

Price Books & Multipliers

If a custom price book uses a multiplier, it will apply the multiplier to every part with a base price. However, if you associate individual parts to a custom price book, the multiplier will NOT be applied to those parts. So, the best practice would be to create a custom price book with a multiplier, and only associate that parts that you DO NOT want to be multiplied.

Sample Use Case

Let's say that you have a Price Book called Test Multiplier (see image below). It has been created to multiply the base price by 2.0.

Price Book Administration List

Select	Order	Expand	Name	Description	Multiplier
	1		Base Price	Default Price Book (1.0 x base price)	Base Price x 1.0
☐	2	⊕	Scouting		Base Price x 1.5
☐	3	⊕	Sports Teams		Base Price x 1.75
☐	4	⊕	Test Multiplier		Base Price x 2.0

So, standard functionality is that if you are using the Test Multiplier price book on the user-side, all parts will be returned with their base price multiplied by 2.0.

Now, let's say that you'd rather not have the part number **MOZZ-STICKS** have its base price multiplied when users select the Test Multiplier price book. You can override this functionality by associating the part with the Test Multiplier price book. You can enter a new price in the field(s) OR you can leave all fields blank and the application will return the price shown in the parentheses (base price).

Price Book Associations

Select	Part Number	Euro (Base Price)	United Kingdom Pound (Base Price)	US Dollar (Base Price)
☐	MOZZ-STICKS			2.50 ($4.00)

NOTES



If users wish to export Price Book Associations, they may do so separately.

Users may still import older Excel documents that contain Price Book Associations columns.



Standard Price Books can accommodate as many currencies as are supported on a BigMachines site.



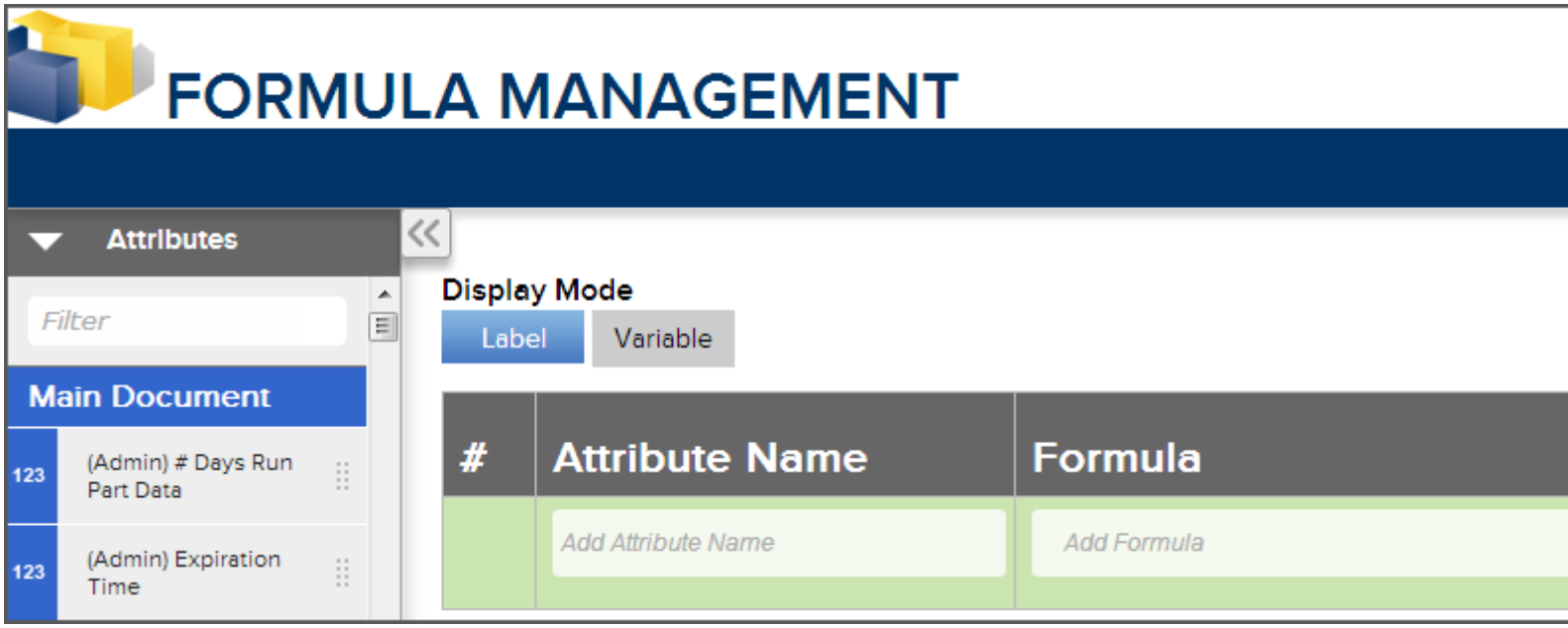
Parts Export no longer includes Price Book Associations columns.

RELATED TOPICS

FORMULA MANAGEMENT OVERVIEW

OVERVIEW

Formula Management allows you to write formulas that will set quote-level and line-level attribute values through a drag & drop interface and visual editor. This enables you to build complex calculations without scripting. *Customer-specific pricing is calculated before formulas are evaluated. An attribute will not have the option to use a formula for its default value until after a formula is created and deployed.*



Formula Management Editor

In the left panel you'll find the following collapsible headers:

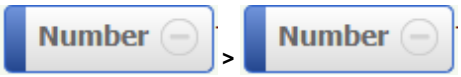
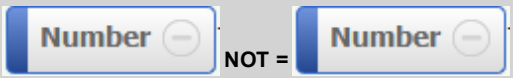
- **Attributes:** Filter or scroll to find main-document and sub-document attributes to set or use in formulas.
- **Functions:** Filter or scroll through a list of functions to be used in formulas.

ADMINISTRATION

You can create formulas to use on quote and line-level attributes by using drag & drop functionality within a visual editor. In the left panel, you can use the filter functionality to quickly search for and find commerce attributes and functions that will be used to create each formula. In the right panel, you will drag & drop an attribute to the attribute name area and then use operators, functions and attributes to create its formula in the associated formula bar.

Operators

Operator	Meaning	Looks Like
+	Addition	

Operator	Meaning	Looks Like
-	Subtraction	
/	Division	
*	Multiplication	
>	Greater Than	
<	Less Than	
=	Equals	
NOT =	Not Equals	
>=	Greater Than or Equal To	
<=	Used in condition grouping	
AND	Used in condition grouping	
OR	Used in condition grouping	
NOT	Used in condition grouping	
()	Used in condition grouping	
LIKE	Used in queries only	

Available Functions

Function	Description	Return Type
sum	Combines a set of numeric inputs and return the value as a number.	Numeric
avg	Calculates the average of a set of numeric inputs and return the value as a number.	Numeric

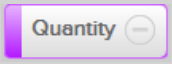
Function	Description	Return Type
sumIf	Evaluates the sum of a given sub-doc attribute of a given attribute based on the condition. The condition will evaluate each sub-document independently.	Numeric
avgIf	Evaluates the average across all sub-documents of a given attribute based on the condition. The condition will evaluate each sub-document independently.	Numeric
capitalizeFirst	Capitalizes the first letter of the string.	String
max	Returns the largest of the given inputs.	Numeric
min	Returns the smallest of the given inputs.	Numeric
median	Evaluates the median of the given numeric inputs.	Numeric
ceil	Converts a number with fraction to the next whole number.	Numeric
round	Returns the first number rounded to the precision defined by the second numeric input.	Numeric
sqrt	Returns the rounded positive square root of the number.	Numeric
pow	Returns the value of the first numeric input raised to the power of the second numeric input.	Numeric
exp	Returns Ruler's number e raised to the power of the numeric input.	Numeric
abs	Returns the positive equivalent to the numeric input entered.	Numeric
mod	Returns the remainder of the first numeric input divided by the second numeric input.	Numeric
formatAsCurrency	Returns a float or integer in the correct currency format.	String
upper	Converts all of the characters in the input string to upper case.	String
lower	Converts all of the characters in the input string to lower case.	String
substring	Returns a part of the text from a larger string.	String
replace	Returns a string with all found occurrences replaced with the next string. The option integer input defines the number of occurrences to replace starting from the beginning of the string.	String
concat	Concatenates, or combines, strings.	String
trim	Removes white space from both ends of a string.	String
len	Returns the length of a string.	Integer
find	Returns the position/index of a substring within a string.	Integer
startswith	Checks whether a string starts with a particular substring.	Boolean
endswith	Checks whether a string ends with a particular substring.	Boolean
isNumber	Returns true when a string is a number and false in all other cases.	Boolean
contains	Checks whether a string contains a particular substring.	Boolean
query	Allows users to search through data tables and return select values.	Column type
toFloat	Converts text that represents a number or integer value into a float.	Float
toString	Converts a numeric value into a string.	String

Function	Description	Return Type
toInteger	Converts a string that represents a number or a float value into an integer.	Integer
if	Checks whether a condition is met and returns one value if TRUE and another value if FALSE.	

Elements

Element	Description
Add Attribute Name	Drag & drop the attribute you want to set into this area. Only one attribute may be set at a time.
Add Formula	Drag & drop attributes and/or functions or type literals and operators into this area. Some functions will have a pop-up box if additional information or parameters are required.
	Click this icon to open a formula for editing.
	Click this icon to delete a formula.
	Click this icon to accept a formula after changes have been made.
	Click this icon to revert to original formula.

Element Colors

Color	Description	Example
Blue	Represents main-document attributes.	
Purple	Represents line item attributes.	
Orange	Represents functions.	
Light Green	Represents literals or static entry.	
Dark Pink	Represents table columns used in a query.	

Auto-Update Checkbox

- Now available for each formula.
- Auto-Update will execute a formula using AJAX. The AJAX execution of formulas occurs before the auto-update on the main document.
- A formula marked for auto-update will run all formulas that share common attributes upon an attribute change.
- Only formulas marked for auto-update will be executed, so be sure to consider which values will be updated and which will not.

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*
        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*
    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



When a + is used in combination with a query function, the query will aggregate the results of the query. This will also occur when a query is used inside of a sum function.

Formulas defined for an attribute can be self-referencing. *Self-referencing formulas only run once.*



Numeric stings are treated as literal (numeric) in conditional formulas.

In setting formulas for string attributes, you will not be able to set a literal value that contains a ". Similarly, the single quote cannot be used as a search value for the LIKE operator of the query function. Neither can be avoided.

The data type of the attribute used in the formula must be the same as the data type of the attribute that is being returned. You can hover over the attribute names to see the variable name and data type; you can hover over functions to see input and return type.

Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items. *When a new line item is added to a quote, the formulas that are defined for line item attributes will run to determine the default values of the attributes.*

Nested functions cannot be used with the 'avgif' and 'sumif' functions in an action; they can be used with conditions.

When using query logic on a data table, if you use an OR operator and the system finds multiple matches, only the first match will be returned.

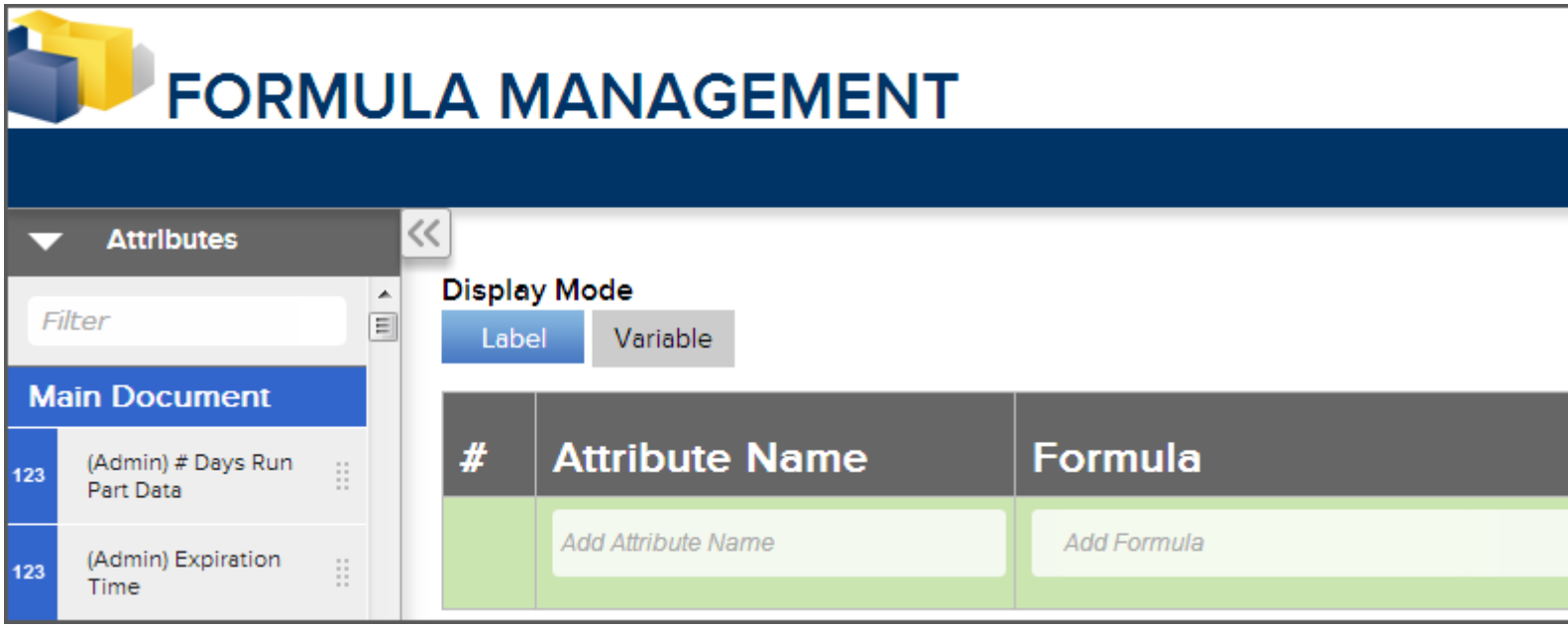
The function 'sumif' cannot return an integer data type.

RELATED TOPICS

FORMULA MANAGEMENT OVERVIEW

OVERVIEW

Formula Management allows you to write formulas that will set quote-level and line-level attribute values through a drag & drop interface and visual editor. This enables you to build complex calculations without scripting. *Customer-specific pricing is calculated before formulas are evaluated. An attribute will not have the option to use a formula for its default value until after a formula is created and deployed.*



Formula Management Editor

In the left panel you'll find the following collapsible headers:

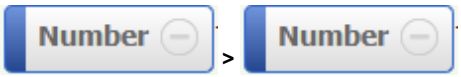
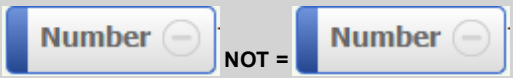
- **Attributes:** Filter or scroll to find main-document and sub-document attributes to set or use in formulas.
- **Functions:** Filter or scroll through a list of functions to be used in formulas.

ADMINISTRATION

You can create formulas to use on quote and line-level attributes by using drag & drop functionality within a visual editor. In the left panel, you can use the filter functionality to quickly search for and find commerce attributes and functions that will be used to create each formula. In the right panel, you will drag & drop an attribute to the attribute name area and then use operators, functions and attributes to create its formula in the associated formula bar.

Operators

Operator	Meaning	Looks Like
+	Addition	

Operator	Meaning	Looks Like
-	Subtraction	
/	Division	
*	Multiplication	
>	Greater Than	
<	Less Than	
=	Equals	
NOT =	Not Equals	
>=	Greater Than or Equal To	
<=	Used in condition grouping	
AND	Used in condition grouping	
OR	Used in condition grouping	
NOT	Used in condition grouping	
()	Used in condition grouping	
LIKE	Used in queries only	

Available Functions

Function	Description	Return Type
sum	Combines a set of numeric inputs and return the value as a number.	Numeric
avg	Calculates the average of a set of numeric inputs and return the value as a number.	Numeric

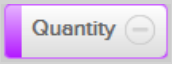
Function	Description	Return Type
sumIf	Evaluates the sum of a given sub-doc attribute of a given attribute based on the condition. The condition will evaluate each sub-document independently.	Numeric
avgIf	Evaluates the average across all sub-documents of a given attribute based on the condition. The condition will evaluate each sub-document independently.	Numeric
capitalizeFirst	Capitalizes the first letter of the string.	String
max	Returns the largest of the given inputs.	Numeric
min	Returns the smallest of the given inputs.	Numeric
median	Evaluates the median of the given numeric inputs.	Numeric
ceil	Converts a number with fraction to the next whole number.	Numeric
round	Returns the first number rounded to the precision defined by the second numeric input.	Numeric
sqrt	Returns the rounded positive square root of the number.	Numeric
pow	Returns the value of the first numeric input raised to the power of the second numeric input.	Numeric
exp	Returns Ruler's number e raised to the power of the numeric input.	Numeric
abs	Returns the positive equivalent to the numeric input entered.	Numeric
mod	Returns the remainder of the first numeric input divided by the second numeric input.	Numeric
formatAsCurrency	Returns a float or integer in the correct currency format.	String
upper	Converts all of the characters in the input string to upper case.	String
lower	Converts all of the characters in the input string to lower case.	String
substring	Returns a part of the text from a larger string.	String
replace	Returns a string with all found occurrences replaced with the next string. The option integer input defines the number of occurrences to replace starting from the beginning of the string.	String
concat	Concatenates, or combines, strings.	String
trim	Removes white space from both ends of a string.	String
len	Returns the length of a string.	Integer
find	Returns the position/index of a substring within a string.	Integer
startswith	Checks whether a string starts with a particular substring.	Boolean
endswith	Checks whether a string ends with a particular substring.	Boolean
isNumber	Returns true when a string is a number and false in all other cases.	Boolean
contains	Checks whether a string contains a particular substring.	Boolean
query	Allows users to search through data tables and return select values.	Column type
toFloat	Converts text that represents a number or integer value into a float.	Float
toString	Converts a numeric value into a string.	String

Function	Description	Return Type
toInteger	Converts a string that represents a number or a float value into an integer.	Integer
if	Checks whether a condition is met and returns one value if TRUE and another value if FALSE.	

Elements

Element	Description
Add Attribute Name	Drag & drop the attribute you want to set into this area. Only one attribute may be set at a time.
Add Formula	Drag & drop attributes and/or functions or type literals and operators into this area. Some functions will have a pop-up box if additional information or parameters are required.
	Click this icon to open a formula for editing.
	Click this icon to delete a formula.
	Click this icon to accept a formula after changes have been made.
	Click this icon to revert to original formula.

Element Colors

Color	Description	Example
Blue	Represents main-document attributes.	
Purple	Represents line item attributes.	
Orange	Represents functions.	
Light Green	Represents literals or static entry.	
Dark Pink	Represents table columns used in a query.	

Auto-Update Checkbox

- Now available for each formula.
- Auto-Update will execute a formula using AJAX. The AJAX execution of formulas occurs before the auto-update on the main document.
- A formula marked for auto-update will run all formulas that share common attributes upon an attribute change.
- Only formulas marked for auto-update will be executed, so be sure to consider which values will be updated and which will not.

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*
        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*
    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



When a + is used in combination with a query function, the query will aggregate the results of the query. This will also occur when a query is used inside of a sum function.

Formulas defined for an attribute can be self-referencing. *Self-referencing formulas only run once.*



Numeric stings are treated as literal (numeric) in conditional formulas.

In setting formulas for string attributes, you will not be able to set a literal value that contains a ". Similarly, the single quote cannot be used as a search value for the LIKE operator of the query function. Neither can be avoided.

The data type of the attribute used in the formula must be the same as the data type of the attribute that is being returned. You can hover over the attribute names to see the variable name and data type; you can hover over functions to see input and return type.

Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items. *When a new line item is added to a quote, the formulas that are defined for line item attributes will run to determine the default values of the attributes.*

Nested functions cannot be used with the 'avgif' and 'sumif' functions in an action; they can be used with conditions.

When using query logic on a data table, if you use an OR operator and the system finds multiple matches, only the first match will be returned.

The function 'sumif' cannot return an integer data type.

RELATED TOPICS

CREATING A FORMULA

OVERVIEW

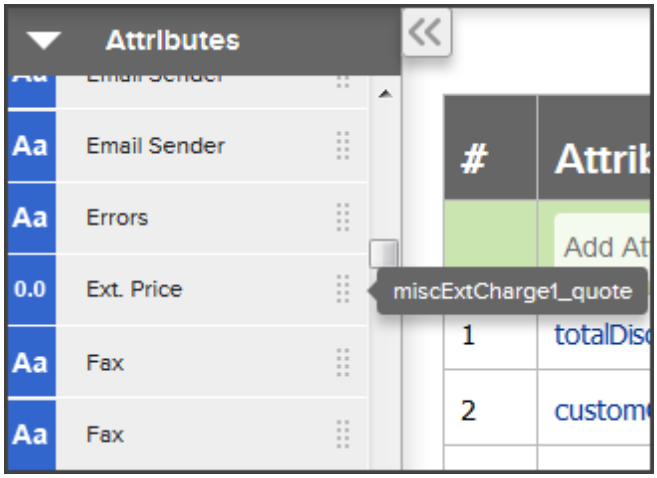
Formulas can be in the main navigation menu for each commerce process. You can create formulas using a drag & drop visual editor, that will set commerce attribute values. You will be able to set both main-document and sub-document level attributes. The use of formula management should reduce the need to use BML scripting.

ADMINISTRATION

Creating a Basic Formula

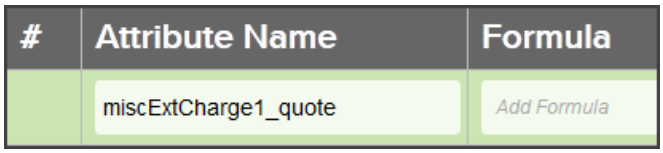
Standard Use Case: You need to set the values of the Extended Price, which is based on Net Price and Quantity.

1. Navigate from: Admin Home Page > Process Definition Quick Links > Select a Process > Click **Formulas**
2. Search for **Extended Price** in the Attribute Filter in the left panel.



When you hover over an attribute, you'll see the variable name and the data type.

3. Drag & drop the Extended Price into the green **Add Attribute** bar under the Attribute Name.



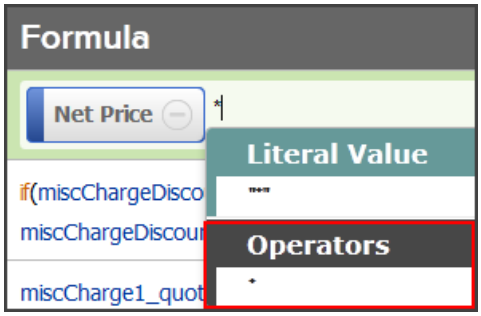
The Attribute Name will display as the variable name, not the label name.

Next, you'll need to populate the formula bar. Here we are using main-document attributes. In this example, Extended Price = Net Price * Quantity

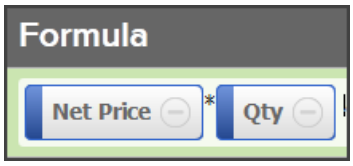
4. Drag & drop the **Net Price** attribute into the formula bar.



5. Type the **multiplication operator** * into the formula bar.



6. Drag & drop the **Quantity** attribute into the formula bar.



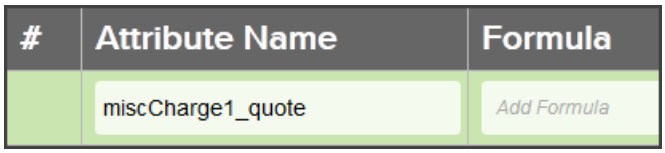
7. Click the icon to add the formula to the formula list.



Creating a Formula Using Functions

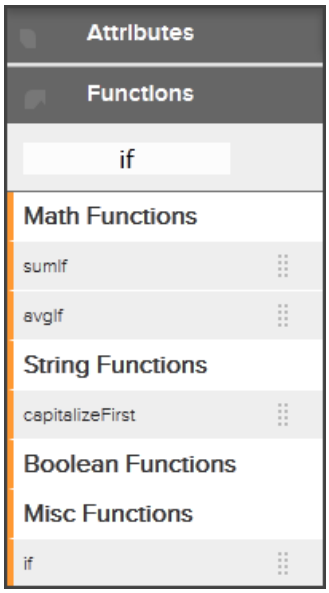
1. Navigate from: Admin Home Page > Process Definition Quick Links > Select a Process > Click **Formulas**
2. Search for **Net Price** in the Attribute Filter in the left panel.

> When you hover over an attribute, you'll see the variable name and the data type.

3. Drag & Drop the Net Price into the green **Add Attribute** bar under the Attribute Name.
4. 

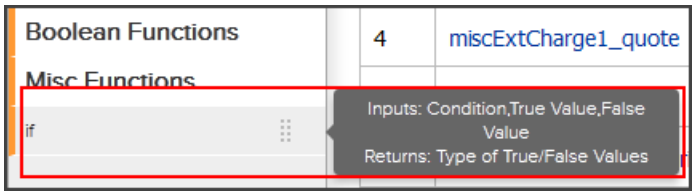
> The Attribute Name will display as the variable name, not the label name.

5. Collapse the **Attributes** section and open the **Functions** section, in the left panel.



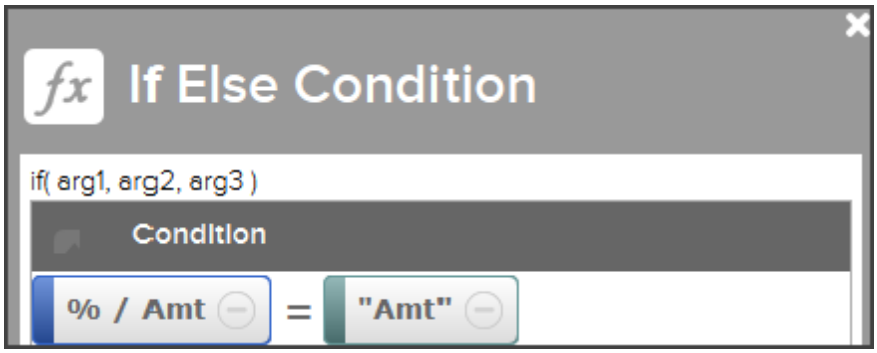
6. Drag & drop the **if function** into the formula bar. *Once the if function has been placed into the formula bar, an if/else condition*

window will pop up.



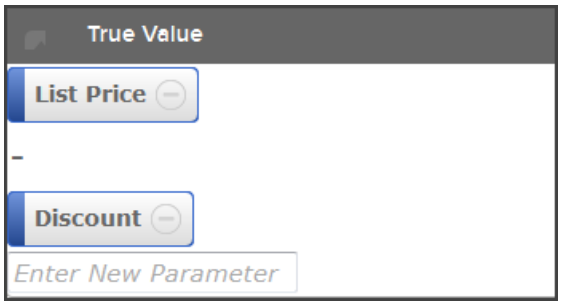
When you hover over a function, you'll see the required inputs and return.

7. Populate the **Condition**. *In this case, we want the Net Price to be calculated if **%/Amt = Amt**.*



Drag & drop is not available in the pop-up windows, so you'll need to begin typing in the label or variable name.

8. Populate the True Value. *This is who the formula will run if the condition is met. In this case, the True Value is **List Price - Discount**.*



9. Populate the **False Value**. *This is who the formula will run if the condition is NOT met.* In this case, the False Value is **Line Price - (Line Price*Discount)/100**.

10. Click **Save**.

11. Click the icon to add the formula to the formula list. *When the Net Price formula is complete, it will appear in the formula list.*



Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*
        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*
    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



Formulas defined for an attribute can be self-referencing. *Self-referencing formulas only run once.*



Numeric strings are treated as literal numbers (numeric) in conditional formulas.

Nested functions cannot be used with 'avgif' and 'sumif' functions in an action; they can be used with conditions.

Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items. *When a new line item is added to a quote, the formulas that are defined for line item attributes will run to determine the default values of the attributes.*

RELATED TOPICS

USING THE QUERY FUNCTION

OVERVIEW

Formula Management allows you to query data tables and return results that will populate a formula and provide a calculation for a line item or main-document attribute.

ADMINISTRATION

Using the Query Function

In this example we have queried a data table to return the current line items volume pricing. *A value of 0.0 will be returned if nothing is found.*

1. Navigate to: Admin Home Page > Process Definition Quick Links> Select a Process > Click **Formulas**
2. Search for **List Price (listPrice_line)** in the Attribute Filter in the left panel.
3. Drag & drop the **Next Price** into the green **Add Attribute** bar under the Attribute Name.

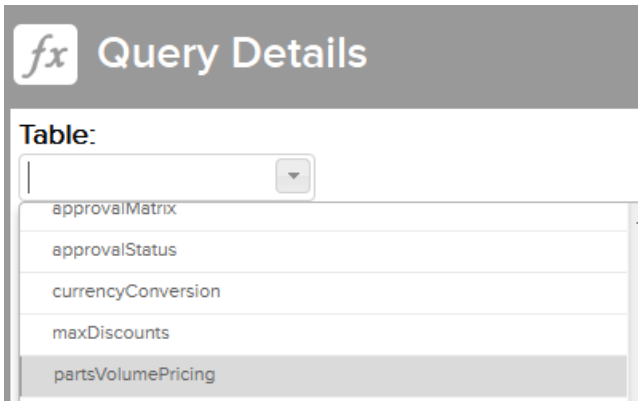
> The Attribute Name will display as the variable name, not the label name.

4. Collapse the **Attributes** section and open the **Functions** section, in the left panel.
5. Drag & drop the query function into the formula bar. *When you hover over a function, you'll see the required inputs and return.*

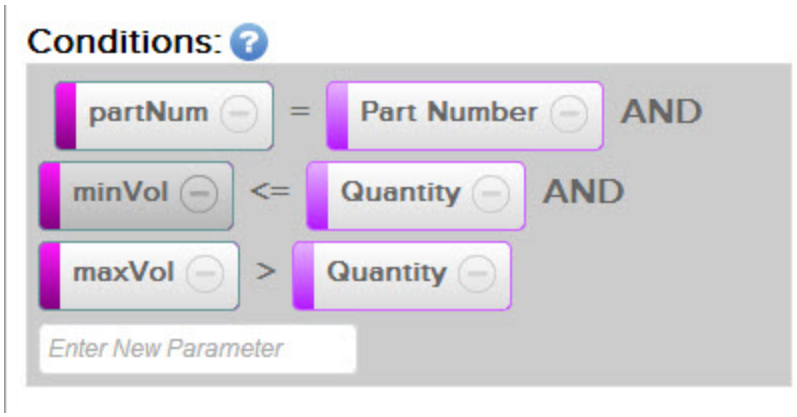


> Once the if function has been placed into the formula bar, a Query Details pop-up window will open.

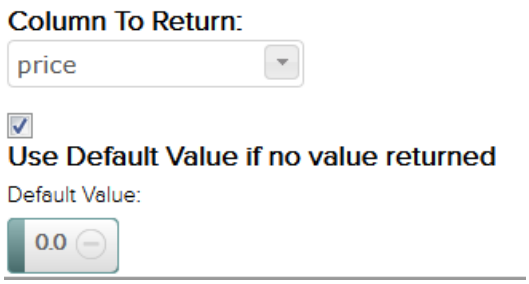
6. Use drop-down menu to choose the data table you will be querying.



7. Enter **Conditions** for the query to run.



8. Select the data table **Column to Return**.
9. Select the checkbox **'Use Default Value if No Value is Entered'**.
10. Enter the **Default Value of 0.0**.



11. Click **Save**.

When the List Price formula is complete, it will appear in the formula list as such:

```
query(partsVolumePricing,price,partNum=_part_numberANDminVol
        <=_price_quantityANDmaxVol>=_price_quantity,0.0)
```

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*
        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*
    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



The data type of attributes used in the formula must be the same as the data type of the attribute that is being returned. When setting a float attribute, you can use integer, float, and/or currency attributes in the formula. However, when setting an integer attribute, the return type must be integer.

Formulas defined for an attribute can be self-referencing. *Self-referencing formulas only run once.*



Formulas will run after the simple modification of an action, but before the advanced BML.

Customer-specific pricing is calculated before formulas are evaluated.

When a + is used in combination with the query function, the query will aggregate the results of the query. This will also occur when a query is used inside of a sum function. When using query logic on a data table, if you can use an OR operator and the system finds multiple matches, only the first match will be returned.

Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items. *When a new line item is added to a quote, the formulas that are defined for line item attributes will run to determine the default values of the attributes.*



Default attribute values can be set to a formula. Attributes that don't have a formula defined, will not have this option on the page.

In setting formulas for string attributes, you will not be able to set a literal value that contains a " (e.g. that"s great). This example will throw an error and there is no way to escape the character. Similarly, the single quote cannot be used as a search value for the LIKE operator of the Query function. This again, cannot be escaped.

RELATED TOPICS

CUSTOMER SPECIFIC PRICING OVERVIEW

OVERVIEW

You can create a specific pricing structure for individual or multiple accounts. fullaccessusers can create pricing rule profiles that contain prices for one or many parts, and if a particular account is tied to that pricing rule, they will see the profile price. This is done so that when a Sales User is quoting a customer that has a pre-determined, contract or pricing agreement, this special pricing should be visible in the UI once the customer and item(s) have been identified. This is a way to define the list price of a given part, based on customer.

Order of Operations

If the same part is present in multiple pricing rules, and within multiple profiles within a rule, the pricing rule price value is derived from the first rule (as show in the list of rules on the administration page) and the first profile within the rule (as shown in the list of profiles section for the rule). You can use drag & drop to modify the order of the rules and profiles relevant for their scenario.

How Prices are Evaluated

The table below explains how the prices are evaluated based on what product type is being added to the transaction.

	Parts	Recommended Part	Model	Recommended Model	Commerce Variable Name
Item Price	Evaluates Price Book Price, then Part Base Price.	Evaluates Price Book Price, then Part Base Price.	Evaluates the Model Base Price, then the Pricing Rule(s), and finally Attribute Value Pricing.	Model Base Price	*_price_item_price_each*
Configurable Price	no value	Recommended Item Price	Item Price	Recommended Item Price	*_price_configurable_price*
List Price	Evaluates User Input, then Item Price.	Evaluates User Input, then Configuration Price, and finally Item Price.	Evaluates User Input, then Item Price.	Evaluates User Input, then Configuration Price, and finally Item Price.	*_price_list_price_each*
Pricing Rule Price	Customer Specific Pricing	Customer Specific Pricing	no value	no value	*_pricing_rule_price_each*
Unit Price	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	*_price_unit_price_each*

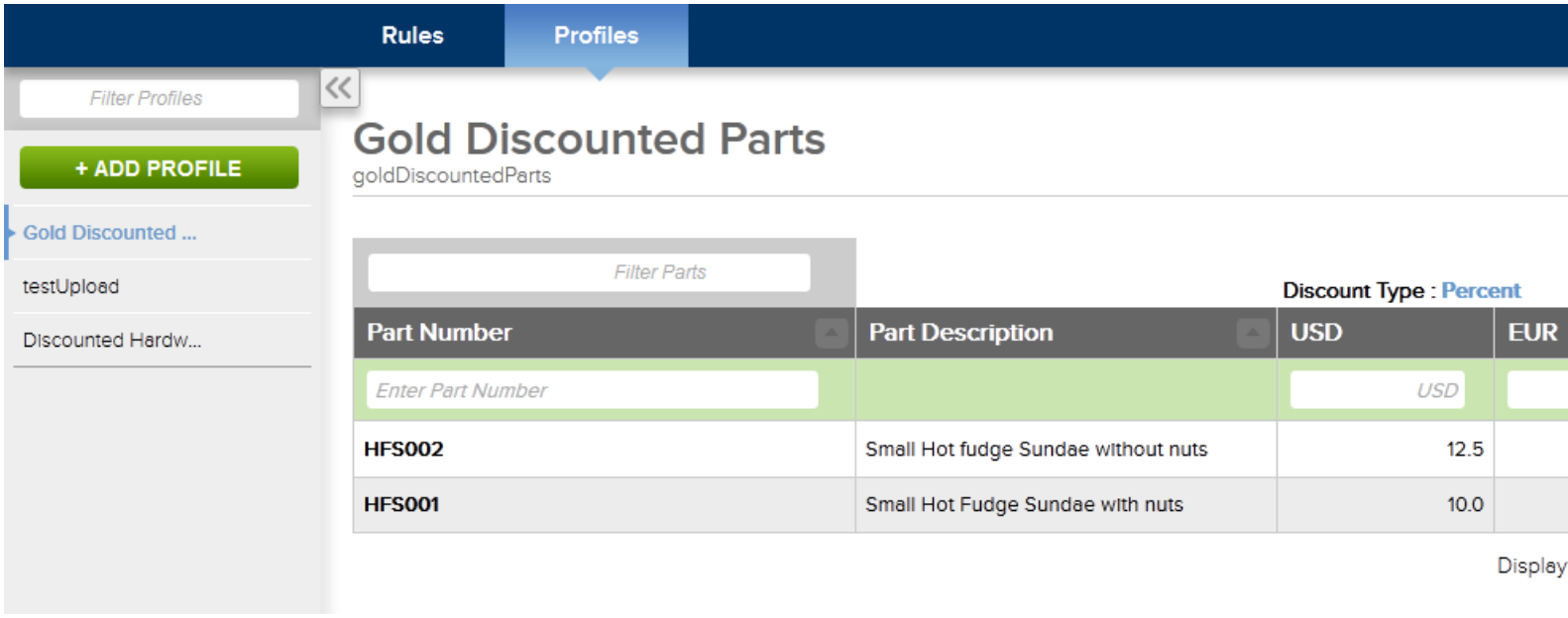


The user input will be evaluated first for List Price and Unit Price ONLY when _price_override_price is set to TRUE.

ADMINISTRATION

Profiles

This page is where you can create a Profile and add parts and pricing.



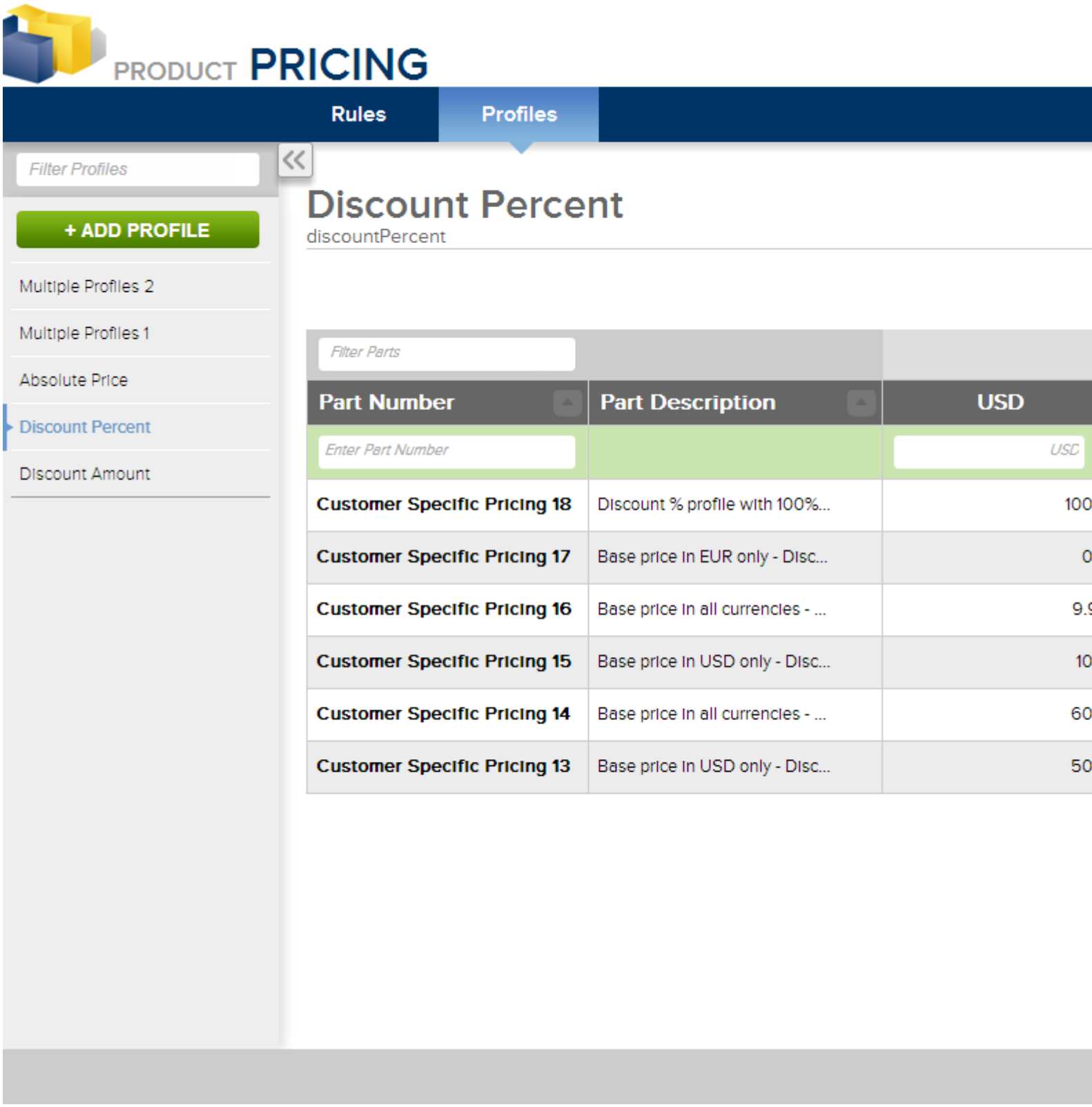
What is it?	What does it do?
Filter Profile	Type in the name of a profile to search through profile names.
+ ADD PROFILE	Click to add a new profile. A pop-up window will appear where you can enter a name, description and discount type.
Filter Parts	Enter characters to filter through parts that have already been added to the profile.
Enter Part Number	Enter part numbers to add them to the profile.
Currency	Enter discount (percentage, amount or absolute price) per each currency.
	Click to add a part to the profile.
Display [] Records	Enter a numeric value that indicates the number of parts you'd like to display on the profile page. You must enter a minimum of 2.
View Linked Rules	Click to view rules that are linked to the profile you are viewing.
Back	Click to return to the Admin Home Page.
	Click to expand the profile page. It will hide the top-side navigation bar.



You must create your Profile before you can create Rules.

Rules

This page is where you can create a Profile and add parts and pricing.



In the right panel, there is also a section where you can add profiles to each rule.

What is it?	What does it do?
Filter Rules	Type in the name of a profile to search through rule names.
+ ADD RULE	Click to add a new rule. A pop-up window will appear where you can enter a name and description.
Filter Account Names	Enter characters to filter through accounts that have already been added to the rule.
Account Name	Enter account name(s) to add to the rule.
Start Date/End Date	Enter start and end dates for each account within the rule.
Status	Select ON/OFF to indicate whether a rule is active or inactive.
	Click to add an account to the rule.
Back	Click to return to the Admin Home Page.
	Click to expand the profile page. It will hide the top-side navigation bar.



You must create your Profile before you can create Rules.

User-Side Functionality

Customer-specific pricing will be visible to users when they invoke a commerce process. The table below depicts the way different prices are applied and how they affect the subtotal.

	Item Price	List Price	Pricing Rule Price	Unit Price	Quantity	Subtotal (List)	Subtotal (Unit)
Direct Buy	$10.00	$10.00	$5.00	$5.00	5	$50.00	$25.00
Parts as Rec Item	$10.00	$8.00	$5.00	$5.00	5	$40.00	$25.00
Part w/o Customer Pricing	$10.00	$8.00	$0.00	$8.00	5	$40.00	$40.00



Account ID must be populated in order to see this pricing.



There have been changes to the way the Subtotal has been derived. Please see **System Attributes** for more information.

Bulk Upload & Download

You can manage "Parts Pricing Rules Information" from Bulk Download/Upload section in Utilities. **There are six files that will be downloaded:**

Files	Description	Notes
PART_PRICES	Contains the price profile variable name, part number and value for all currencies.	This file should be used when new parts need to be added to the profile.
PRICE_PROFILES	Contains price profile details and the type.	This file should be used to add/update price profiles.
PRICING_RULES	Contains the definition of the pricing rules.	This file should be used to add/update basic pricing rule information (name, variable name and description).
PRICING_RULES_ CUSTOMERS	Associates customer(s) with pricing rules and dates.	The dates are broken into separate columns with the format defined in brackets in the column header.
PRICING_RULES_ PROFILES	Associates profiles with pricing rules.	N/A
upload.properties	Always carries the same text and is required for uploads.	Out of the other five files, only the file that has changes should be uploaded with this file.



This can only be done through .CSV and .ZIP files.

Column Pagination

- When 10 or more currencies are being used, fields will be paginated into columns to eliminate horizontal scrolling.
    - Pagination will only occur when five or more currencies are being utilized in: AVP Entry Manager, Pricebook Associations, and Customer Specific Pricing.
    - Paginated columns will display 5 at a time, each identified by its currency code.
    - Click the left/right arrows to load the next/previous set of columns.
- The “Currency: Show” drop-down menu allows users to access a currency (or set of currencies) directly without having to navigate through the paginated columns.



- In Customer Specific Pricing, any changes that a user makes on a page with pagination will be saved implicitly when switching column views (user must stay within area for the implicit changes to take place).
- In any admin pages involving currency, when 10 or more currencies are being used, currencies will be displayed in a five-column layout to reduce vertical scrolling.

- **User side:** the currency code will be displayed, followed by the display name.
- **fullaccessuser side:** the base currency will be highlighted for easy recognition, and precede all other currencies used. When admin hovers over the currency code, a hover tooltip will display the currency name.

Best Practices

The following will help ensure that your pricing function does not become too large.

- Any added/changed functionality should be broken into logical modules for each pricing requirement, not all placed into one large add-on function. **Example:** if a requirement for CSP and a requirement for promotional pricing exist, they should each have their own function. *This keeps the code organized and manageable.*
- Any pricing functionality that is added to the Base Template/standard functionality should be addressed as follows:
  a. Create a new BML Library for each module; this is where the BML should be written for the new functionality.
  b. Call the BML Library from the pricing function so that it passes through the necessary parameters (e.g. Base Template/standard Max Discount functionality). *The BML Library call should be assigned to a variable, causing the return value of the BML Library to be stored in that variable.*
- When changing Base Template/standard functionality, changes must occur directly in the pricing function or placed in a BML Library and called from the pricing function using the following guidelines:
  a. **Small changes:** Place changes directly into the pricing function. *Existing lines should never be deleted (only commented out) and comments detailing the change should be added.*
  b. **Large changes:** Create a BML Library for the new functionality and have the call occur from within the pricing function (e.g. Base Template/standard Volume Pricing functionality).
- Do not put validations in the pricing script; place them in the validations section or a separate library.
- Eliminate unused code.
- Don't declare variables unnecessarily. *If a variable is used as a static value, do not declare a variable for this purpose.*
- If multiple loops are used to iterate through line items, combine them (if possible). *This will reduce compile size and improve buy-side performance.*
- Combine conditional blocks of text. If several blocks of text are controlled by the same condition (e.g. "if" statements), combine them so that the compiler must only assess the condition once.

NOTES



Hover tooltip has been added to the admin side to display the full name of a currency code.

fullaccessusers can set a customer-specific price of '0' for parts.

If the profile type is an Absolute amount, define amount values in USD and EURO. If there is a EURO specific price, you should populate it with the price-book price.

If the profile type is Discount Amount, set 0 as the discount amount if no CSP exists in a specific currency. Do the same for a Discount Percent profile type.



Pricing rules will only be shown in commerce once an account has been synced; Account ID must be populated.

CSP supersedes Recommended Item Price.

We recommend that values in the price profile be populated for all supported currencies for a given part. **Example:** if your application supports USD and EURO, when a part is added to a profile, a price/value should be defined for both the USD and EURO currencies.



Error messages are not clear in bulk upload/download. Error message will currently not tell you why records have failed. Errors can be viewed in the application Error Logs.

We are not currently supporting Siebel Integration.

RELATED TOPICS

CUSTOMER SPECIFIC PRICING OVERVIEW

OVERVIEW

You can create a specific pricing structure for individual or multiple accounts. fullaccessusers can create pricing rule profiles that contain prices for one or many parts, and if a particular account is tied to that pricing rule, they will see the profile price. This is done so that when a Sales User is quoting a customer that has a pre-determined, contract or pricing agreement, this special pricing should be visible in the UI once the customer and item(s) have been identified. This is a way to define the list price of a given part, based on customer.

Order of Operations

If the same part is present in multiple pricing rules, and within multiple profiles within a rule, the pricing rule price value is derived from the first rule (as show in the list of rules on the administration page) and the first profile within the rule (as shown in the list of profiles section for the rule). You can use drag & drop to modify the order of the rules and profiles relevant for their scenario.

How Prices are Evaluated

The table below explains how the prices are evaluated based on what product type is being added to the transaction.

	Parts	Recommended Part	Model	Recommended Model	Commerce Variable Name
Item Price	Evaluates Price Book Price, then Part Base Price.	Evaluates Price Book Price, then Part Base Price.	Evaluates the Model Base Price, then the Pricing Rule(s), and finally Attribute Value Pricing.	Model Base Price	*_price_item_price_each*
Configurable Price	no value	Recommended Item Price	Item Price	Recommended Item Price	*_price_configurable_price*
List Price	Evaluates User Input, then Item Price.	Evaluates User Input, then Configuration Price, and finally Item Price.	Evaluates User Input, then Item Price.	Evaluates User Input, then Configuration Price, and finally Item Price.	*_price_list_price_each*
Pricing Rule Price	Customer Specific Pricing	Customer Specific Pricing	no value	no value	*_pricing_rule_price_each*
Unit Price	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	Evaluates User Input, then Customer Specific Pricing, and finally List Price.	*_price_unit_price_each*

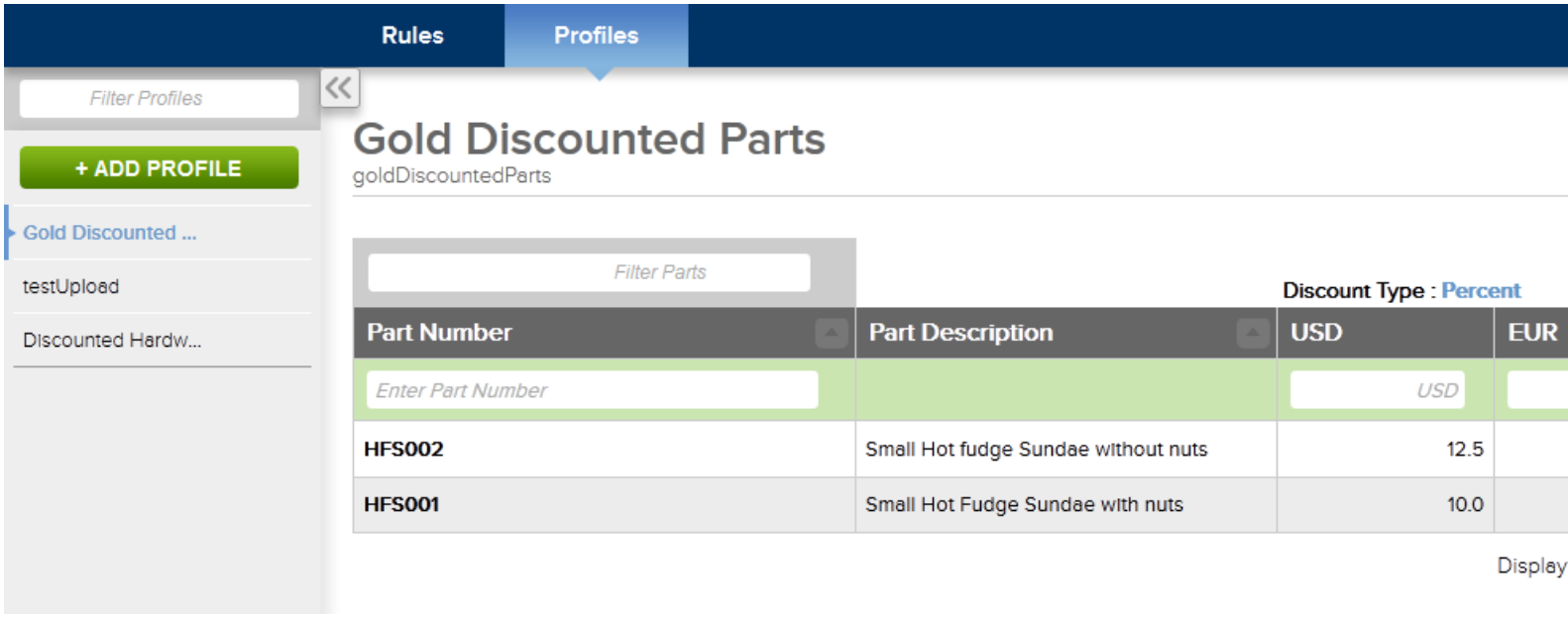


The user input will be evaluated first for List Price and Unit Price ONLY when _price_override_price is set to TRUE.

ADMINISTRATION

Profiles

This page is where you can create a Profile and add parts and pricing.



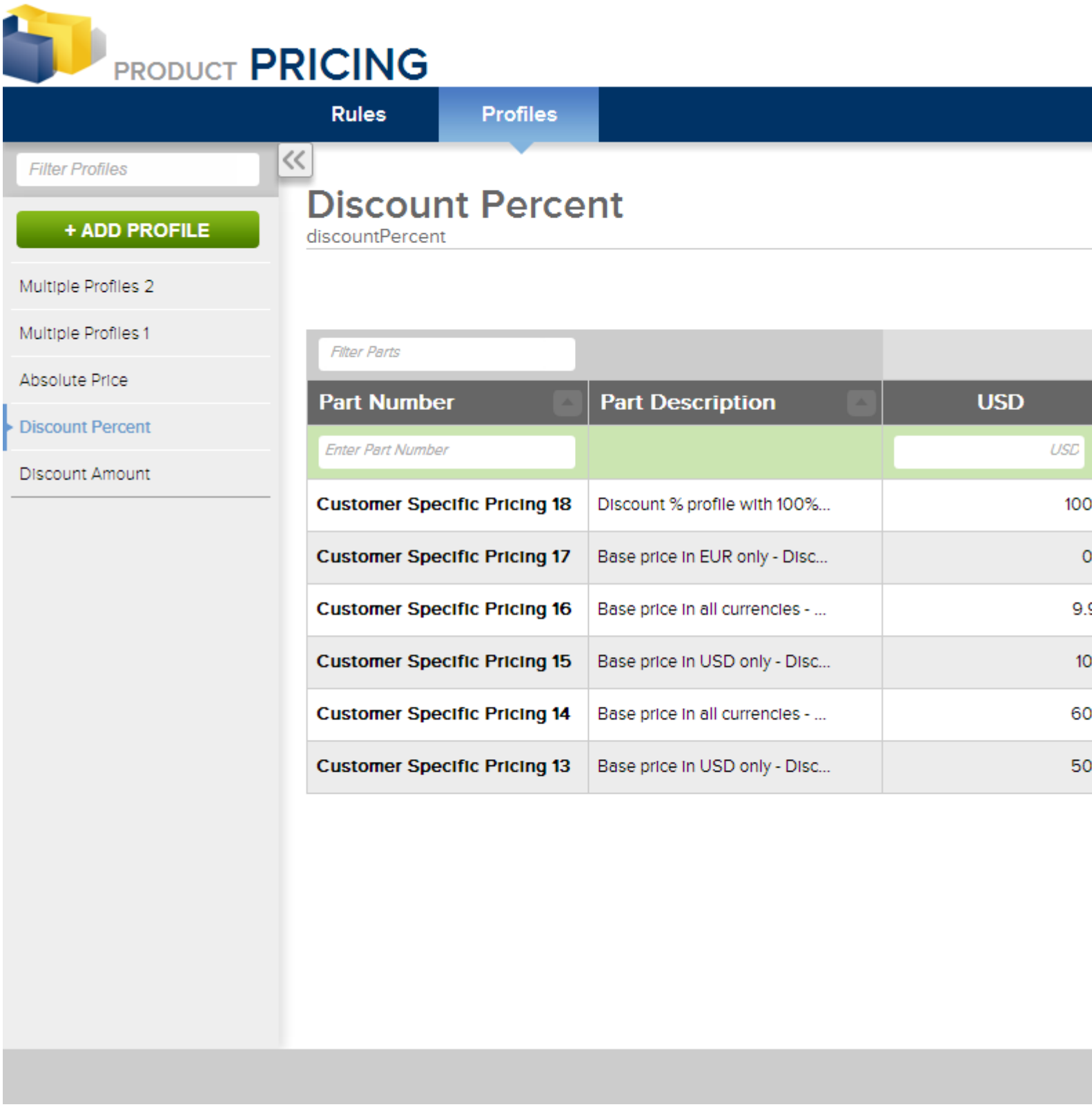
What is it?	What does it do?
Filter Profile	Type in the name of a profile to search through profile names.
+ ADD PROFILE	Click to add a new profile. A pop-up window will appear where you can enter a name, description and discount type.
Filter Parts	Enter characters to filter through parts that have already been added to the profile.
Enter Part Number	Enter part numbers to add them to the profile.
Currency	Enter discount (percentage, amount or absolute price) per each currency.
	Click to add a part to the profile.
Display [] Records	Enter a numeric value that indicates the number of parts you'd like to display on the profile page. You must enter a minimum of 2.
View Linked Rules	Click to view rules that are linked to the profile you are viewing.
Back	Click to return to the Admin Home Page.
	Click to expand the profile page. It will hide the top-side navigation bar.



You must create your Profile before you can create Rules.

Rules

This page is where you can create a Profile and add parts and pricing.



In the right panel, there is also a section where you can add profiles to each rule.

What is it?	What does it do?
Filter Rules	Type in the name of a profile to search through rule names.
+ ADD RULE	Click to add a new rule. A pop-up window will appear where you can enter a name and description.
Filter Account Names	Enter characters to filter through accounts that have already been added to the rule.
Account Name	Enter account name(s) to add to the rule.
Start Date/End Date	Enter start and end dates for each account within the rule.
Status	Select ON/OFF to indicate whether a rule is active or inactive.
	Click to add an account to the rule.
Back	Click to return to the Admin Home Page.
	Click to expand the profile page. It will hide the top-side navigation bar.



You must create your Profile before you can create Rules.

User-Side Functionality

Customer-specific pricing will be visible to users when they invoke a commerce process. The table below depicts the way different prices are applied and how they affect the subtotal.

	Item Price	List Price	Pricing Rule Price	Unit Price	Quantity	Subtotal (List)	Subtotal (Unit)
Direct Buy	$10.00	$10.00	$5.00	$5.00	5	$50.00	$25.00
Parts as Rec Item	$10.00	$8.00	$5.00	$5.00	5	$40.00	$25.00
Part w/o Customer Pricing	$10.00	$8.00	$0.00	$8.00	5	$40.00	$40.00



Account ID must be populated in order to see this pricing.



There have been changes to the way the Subtotal has been derived. Please see **System Attributes** for more information.

Bulk Upload & Download

You can manage "Parts Pricing Rules Information" from Bulk Download/Upload section in Utilities. **There are six files that will be downloaded:**

Files	Description	Notes
PART_PRICES	Contains the price profile variable name, part number and value for all currencies.	This file should be used when new parts need to be added to the profile.
PRICE_PROFILES	Contains price profile details and the type.	This file should be used to add/update price profiles.
PRICING_RULES	Contains the definition of the pricing rules.	This file should be used to add/update basic pricing rule information (name, variable name and description).
PRICING_RULES_ CUSTOMERS	Associates customer(s) with pricing rules and dates.	The dates are broken into separate columns with the format defined in brackets in the column header.
PRICING_RULES_ PROFILES	Associates profiles with pricing rules.	N/A
upload.properties	Always carries the same text and is required for uploads.	Out of the other five files, only the file that has changes should be uploaded with this file.



This can only be done through .CSV and .ZIP files.

Column Pagination

- When 10 or more currencies are being used, fields will be paginated into columns to eliminate horizontal scrolling.
    - Pagination will only occur when five or more currencies are being utilized in: AVP Entry Manager, Pricebook Associations, and Customer Specific Pricing.
    - Paginated columns will display 5 at a time, each identified by its currency code.
    - Click the left/right arrows to load the next/previous set of columns.
- The “Currency: Show” drop-down menu allows users to access a currency (or set of currencies) directly without having to navigate through the paginated columns.



- In Customer Specific Pricing, any changes that a user makes on a page with pagination will be saved implicitly when switching column views (user must stay within area for the implicit changes to take place).
- In any admin pages involving currency, when 10 or more currencies are being used, currencies will be displayed in a five-column layout to reduce vertical scrolling.

- **User side:** the currency code will be displayed, followed by the display name.
- **fullaccessuser side:** the base currency will be highlighted for easy recognition, and precede all other currencies used. When admin hovers over the currency code, a hover tooltip will display the currency name.

Best Practices

The following will help ensure that your pricing function does not become too large.

- Any added/changed functionality should be broken into logical modules for each pricing requirement, not all placed into one large add-on function. **Example:** if a requirement for CSP and a requirement for promotional pricing exist, they should each have their own function. *This keeps the code organized and manageable.*
- Any pricing functionality that is added to the Base Template/standard functionality should be addressed as follows:
  a. Create a new BML Library for each module; this is where the BML should be written for the new functionality.
  b. Call the BML Library from the pricing function so that it passes through the necessary parameters (e.g. Base Template/standard Max Discount functionality). *The BML Library call should be assigned to a variable, causing the return value of the BML Library to be stored in that variable.*
- When changing Base Template/standard functionality, changes must occur directly in the pricing function or placed in a BML Library and called from the pricing function using the following guidelines:
  a. **Small changes:** Place changes directly into the pricing function. *Existing lines should never be deleted (only commented out) and comments detailing the change should be added.*
  b. **Large changes:** Create a BML Library for the new functionality and have the call occur from within the pricing function (e.g. Base Template/standard Volume Pricing functionality).
- Do not put validations in the pricing script; place them in the validations section or a separate library.
- Eliminate unused code.
- Don't declare variables unnecessarily. *If a variable is used as a static value, do not declare a variable for this purpose.*
- If multiple loops are used to iterate through line items, combine them (if possible). *This will reduce compile size and improve buy-side performance.*
- Combine conditional blocks of text. If several blocks of text are controlled by the same condition (e.g. "if" statements), combine them so that the compiler must only assess the condition once.

NOTES



Hover tooltip has been added to the admin side to display the full name of a currency code.

fullaccessusers can set a customer-specific price of '0' for parts.

If the profile type is an Absolute amount, define amount values in USD and EURO. If there is a EURO specific price, you should populate it with the price-book price.

If the profile type is Discount Amount, set 0 as the discount amount if no CSP exists in a specific currency. Do the same for a Discount Percent profile type.



Pricing rules will only be shown in commerce once an account has been synced; Account ID must be populated.

CSP supersedes Recommended Item Price.

We recommend that values in the price profile be populated for all supported currencies for a given part. **Example:** if your application supports USD and EURO, when a part is added to a profile, a price/value should be defined for both the USD and EURO currencies.



Error messages are not clear in bulk upload/download. Error message will currently not tell you why records have failed. Errors can be viewed in the application Error Logs.

We are not currently supporting Siebel Integration.

RELATED TOPICS

ADDING & EDITING CSP PROFILES & RULES

OVERVIEW

You can create a specific pricing structure for individual or multiple accounts. fullaccessusers can create pricing rule profiles that contain prices for one or many parts, and if a particular account is tied to that pricing rule, they will see the profile price. This is done so that when a Sales User is quoting a customer that has a pre-determined, contract or pricing agreement, this special pricing should be visible in the UI once the customer and item(s) have been identified. This is a way to define the list price of a given part, based on customer.

ADMINISTRATION

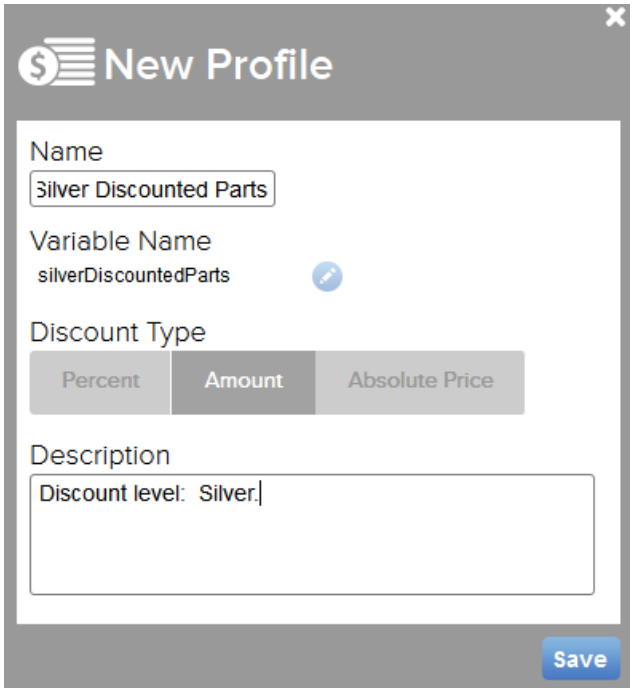
Adding a CSP Profile

1. Navigate from Admin Home Page > Products > **Pricing**.
2. Click on the **Profiles** tab.
3. Click the 

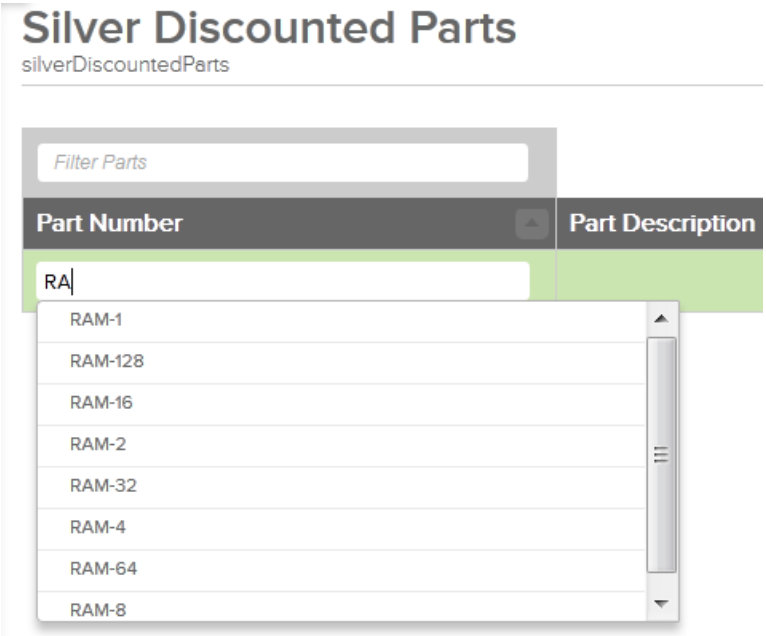
button in the left panel. *A pop-up window will appear.*
4. Enter the **Name**. *The variable name will auto-populate.*
5. Select the **Discount Type**:

- **Percent:** Select if you will enter discounts in percentages. Enter '0' if no customer-specific pricing exists for a specific currency.
- **Amount:** Select if you will enter discounts in amount. Enter '0' if no customer-specific pricing exists for a specific currency.
- **Absolute Price:** Select if you will enter a static price, not discount, for each currency.

6. Enter a **Description**, if desired.
7. Click **Save**.



8. Click within the empty box under **Part Number** and begin to populate with a part number. *Once you begin typing, the available parts will begin to filter.*



9. Populate discount for each currency. Please see **step 5** for a description of discount types.

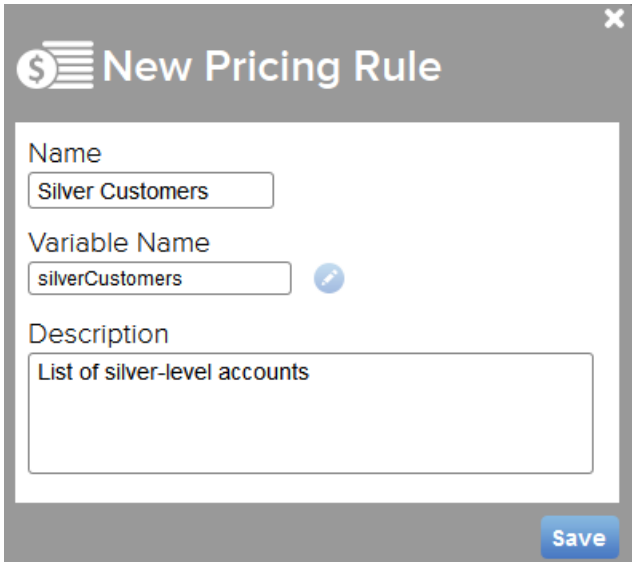
10. Click  when you've finished populating the discount or absolute price for each currency.

Part Number	Part Description	USD	EUR	GBP	AUD
Enter Part Number		USD	EUR	GBP	AUD
RAM-128	128 GB RAM 1333MHz	25	10	0	0

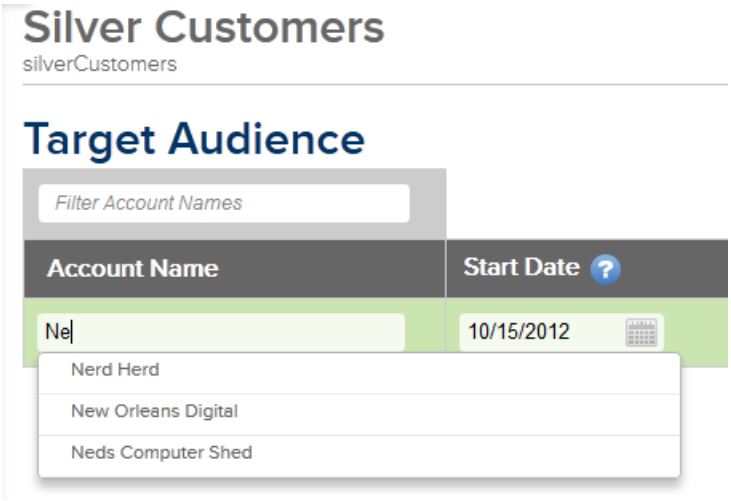
11. Repeat **steps 9-10** for each part that will be added to the profile.

Adding a CSP Rule

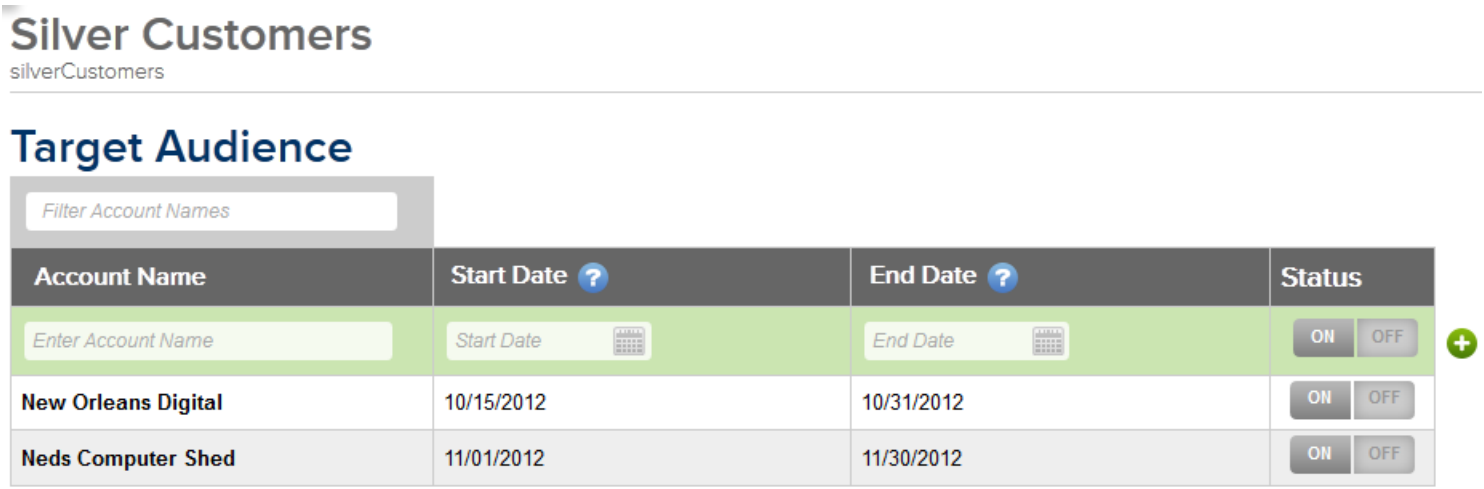
1. Navigate from Admin Home Page > Products > **Pricing**.
2. Click on the **Rules** tab.
3. Click the + ADD RULE button in the left panel. *A pop-up window will appear.*
4. Enter the **Name** and **Description** (if desired). *The variable name will auto-populate.*
5. Click **Save**.



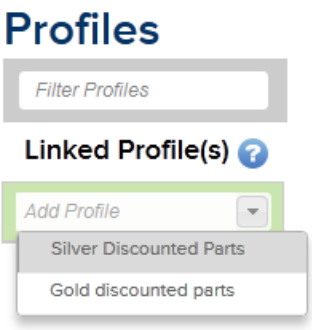
6. Click within the empty box under **Account Name** and begin to populate with a part number. *Once you begin typing, the available accounts will begin to filter.*



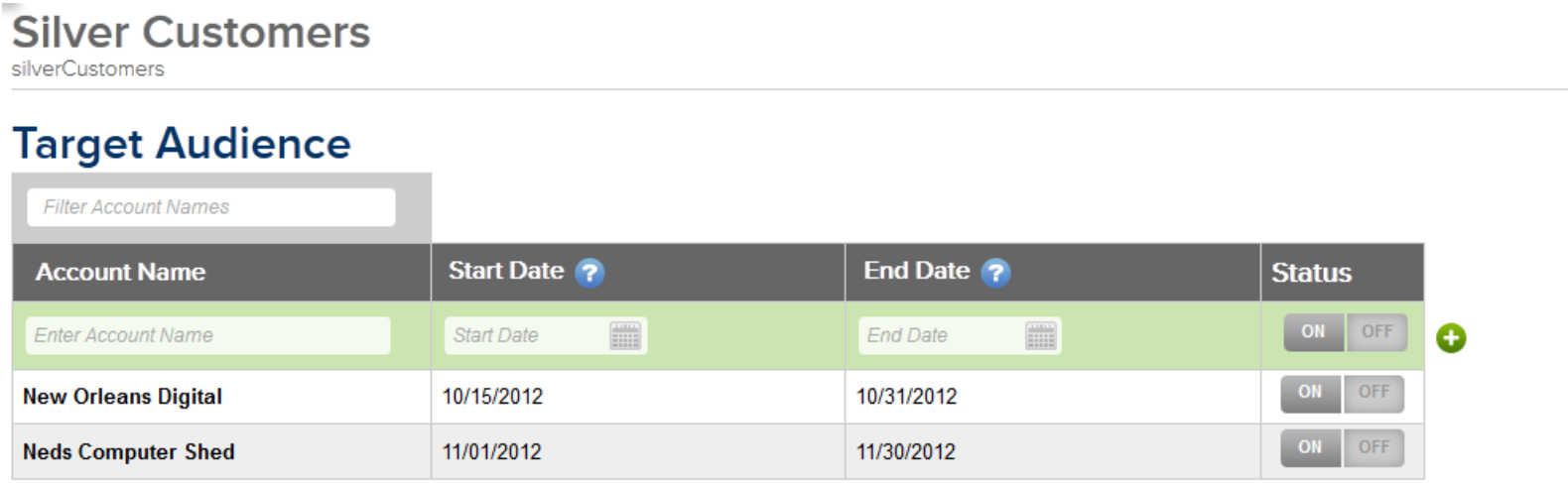
7. Enter **Start and End Dates** for each account in the rule.
8. Select **ON** or **OFF** to indicate whether the rules should be active or inactive for the account.
9. Click  when you've finished to add the account.
10. Repeat **steps 6-9** for each account that will be added to the rule.



11. Link a profile, to the rule you created, from the right panel.

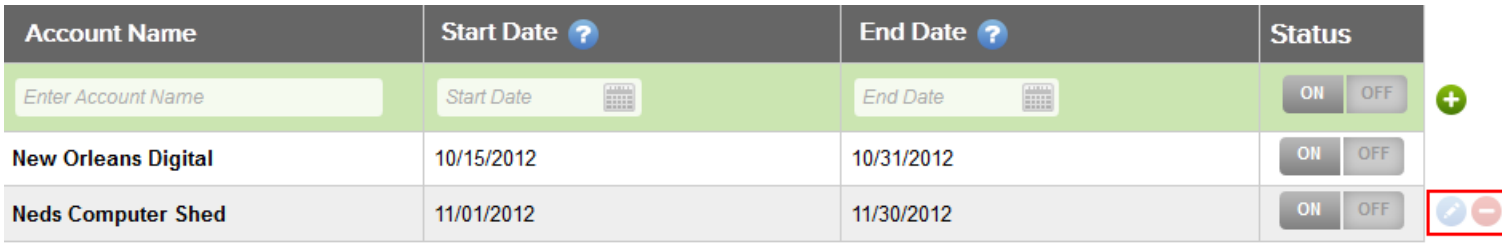


FINAL RESULT:

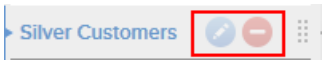


Editing or Deleting Rules & Accounts

1. Hover over the account, just past the Status bar.



2. Click the edit or delete icon.
3. Hover over the rule name and edit or delete the rule.



You cannot delete a CSP rule until you've removed all linked profiles and deleted all accounts.

NOTES

If the profile type is an Absolute amount, define amount values in USD and EURO. If there is a EURO specific price, you should populate it with the price-book price.

If the profile type is Discount Amount, set 0 as the discount amount if no CSP exists in a specific currency. Do the same for a Discount Percent profile type.

We recommend that values in the price profile be populated for all supported currencies for a given part. **Example:** if your application supports USD and EURO, when a part is added to a profile, a price/value should be defined for both the USD and EURO currencies.

You must create a Profile before Rules can be created.

RELATED TOPICS

COMMERCE PROCESS OVERVIEW

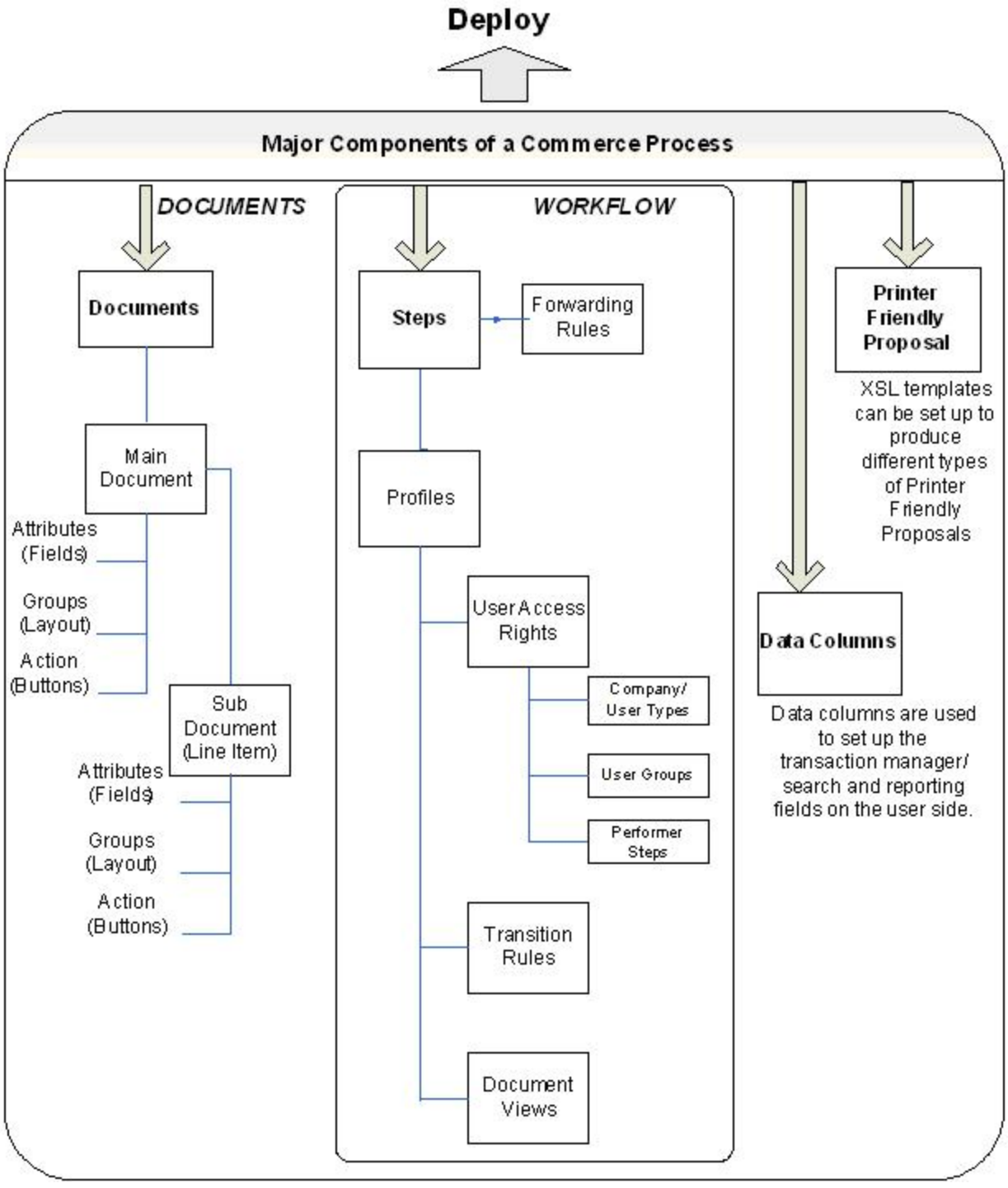
OVERVIEW

Commerce processes are used to create templates for the purchasing processes used by your company. Here, you set up your quoting, ordering, approval, and other workflow processes. By creating ordered sets of commerce document templates, associated attributes and actions, you can enable buyers and supplier agents to conduct commerce on your system. Built from the ground up, each commerce process can be customized according to how you want to structure data flow between users during transactions. You can create one process for all commerce transactions or multiple processes based on different purchase types.

- **Process Definition:** Define your commerce documents, attributes, actions and proposal views.
- **Process Invocation:** Define rules that enable the user to invoke your commerce process from various pages in the application including configuration and parts pages.

All commerce process tasks are administrated in an inactive, pending state. In order for a new process (or a change to an existing process) to be visible on the production side of the system, the process must be deployed. After a commerce process is deployed, certain commerce process administration functionality becomes hidden from view and only administrable by an implementation engineer. This happens because deployed processes contain user data once in use by partner organizations and host companies. In order to preserve user data in live commerce processes, some types of changes can only be performed on cloned processes that are then migrated into the system. At this time, the commerce migration process can only be performed by a BigMachines implementation engineer.

Commerce Process Administration Infographic



Commerce Stylesheet Editor

Stylesheets are basically CSS files that are used to set up commerce processes. This stylesheet provides an easy interface for the full-accessuser to define commerce transactions with a choice of fonts and icons. A default CSS is provided that the user can customize using the Commerce StyleSheet Editor page. **Specify Attribute Properties for:**

1. Click **Browse** to open the File Manager.
2. Scroll through the images in the File Manager. *To select an image, click the image thumbprint. This displays the selected icon next to Browse.*
3. Click **Reload** to make the change permanent. *Ensure that the image is of the size 20 pixel by 20 pixel.*
4. Click **Update** to save the changes.
5. Click **Back** to return to the previous page.

Attribute Properties	Description
Font Color	Specify the text color using the hexadecimal format. For a preview of the color, click Update<. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Font Size	Select the font-size using the drop-down list provided. You can also specify the font size in either points or pixels. Once decided, click Update to make the change permanent.
Font Style	Select the font-style using the drop-down list provided. Once decided, click Update to make the change permanent.
Font Weight	Select the font-weight using the drop-down list provided. Once decided, click Update to make the change permanent.
Font Family	Select the font-family using the drop-down list provided. Once decided upon the family, click Update to make the change permanent. *The default font is Tahoma.*

Upload icons for :

- Group Expand
- Group Collapse
- Line Item

Auto-Update Fields

Instantly update fields in the transaction main document based on trigger fields on the same document. The typical use cases might include Instant Pricing in the line item section based on changes to quantity of line items. Additionally, the admin now has the ability to display column totals for line item columns. These totals are also updated automatically.

1. This feature allows the fullaccessusers to set up calculations to update the fields on the main document when trigger fields associated with it are modified.
    - The fullaccessuser enables auto update on the transaction by checking the trigger property on attributes. *When these attributes are changed by the end user, the auto-update function is instantly run, which updates fields in the main document.*
    - The trigger attributes can be of menu, numeric, currency or boolean type and can include both main document and sub-document attributes.

- The auto-update function can update main document attributes or line item attributes that are displayed on the main document. *Only currency, integer, and text fields can be modified by the Auto-Update script.*
- The fullaccessuser must also define an auto-update function at the main document level that returns the affected attributes due to changes in the trigger fields. The return format of this rule is: `documentNumber~variableName~value [ |documentNumber~variableName~value ]*`
- Update Line Item function is not run automatically when the auto-update occurs. *The user must invoke a modify type action for the changes to be saved to the database.*

2. A feature in the line item column group allows the fullaccessuser to set up column totals.
   - When the auto-update rule fires, in addition to updating the pricing fields in the line item section (if applicable) it also updates the sum without clicking on any button.
   - A line item of numeric or currency type can be auto-summed. *This feature is not available for attributes of other types.*
3. All auto-updated fields are highlighted in sharp yellow and fade away to a dull yellow to indicate that the fields have been modified. *The dull yellow is also an indicator that the modified values have not been updated into the database until an appropriate "modify" action is clicked.*

Managing a Commerce Process

This page displays a list of the commerce processes currently available along with various Navigation options, Quick Links for the other available options, descriptions for the available processes, and the date and time the processes were Last Deployed.

Available Action	Description
Delete a Process	Select all or a required Process Name that is to be deleted using the Selection Option and click Delete to delete it.
Edit Details for Specific Processes	Click on the Process Name link to go to the Process Administration Page.
Navigate to Process-Related Sections	Select an option from the Navigation drop-down list and click List. This displays the corresponding screens for the option selected.
Quick Links	This opens a new window displaying a list of process that can be deployed together using Deploy. Click Refresh on the window to get the current data. Click Close to close it.
Reporting	This displays the Reporting Setup page for managing reports generation. This option is available if the advanced reporting module is turned on for your company.
Migration	his button is visible based on a commerce global setting under Admin Home Page > Global System Settings and Utilities > Commerce.
Add	Goes to the Process Administration page and add details for the commerce process.
Apply	Saves the changes made on the Process Administration Page.
Cancel	Returns to the Processes page without saving your changes.

Options available on the Navigation drop-down List:

- Document: Clicking List next to Documents opens the Documents List page for managing documents within the select process.

- Steps: Clicking List next to Steps displays the Steps list for managing workflows. *At least one main document must be created before Steps can be created and accessed.*
- Process Actions: Clicking List next to Process Actions displays the Process Actions List page for customizing process actions. *At least one main document must be created before Process Actions can be created and accessed.*
- Data Columns: Clicking LIst next to Data Columns displays the Mapped Data Columns page for managing data columns. *At least one main document must be created before Data Columns can be created and accessed.*
- Process Manager: Clicking LIst next to Process Manager displays the Process Manager Columns page for mapping the process columns for the Commerce Manager. *At least one main document must be created before Process Manager can be created and accessed.*
- Printer Friendly/History XSL Views
- Composite Attribute XSL Views
- Integration XSL Files
- Deployment Center



Integration is ONLY available for companies that his this module turned on. If you want to learn more about BigMachines integration module, please contact your system fullaccessuser.

ADMINISTRATION

Setting up a Commerce Process

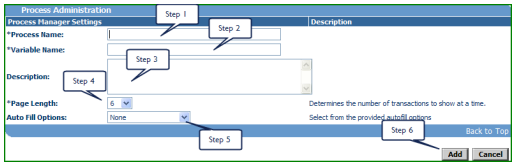
1. Create a process and save it.
2. Add main document(s) and sub document(s) to the commerce process.
3. For each document, create attributes, actions, and views.
4. Develop XSL views for print, email, and change history actions.
5. Customize process actions for the Transaction Manager.
6. Create the Transaction Manager Column layout using Data Columns.
7. Design the process workflow steps. *In each workflow step, create participant profiles to control user access. Participant profiles must be further defined using:*

- User access rights
- Transition rules
- Document views

8. Set up forwarding rules within workflow steps to automatically forward transactions to groups upon creating or saving a transaction.
9. Create timers to automatically perform document actions and auto-forwarding rules to support collaborative workflows.
10. Deploy the commerce process.

Adding a Commerce Process

1. Enter a **Process Name** in the field provided. The process name you choose is used as the navigation link to the Commerce Manager. *This is a required field.*
2. Enter a unique **Variable Name** and a **Description**. *This is a required field.*
3. Enter a description for the process being added.



4. Choose the number of commerce transactions you want to display on the Commerce Manager page using the **Page Length** drop-down menu. *This is a required field.*
5. Choose an **Auto Fill Option** from the drop-down menu if you want new commerce documents auto-populated with user account data or Accounts Information. *You can also opt for no auto fill by selecting None.*
6. Click **Add**. *This adds the new process and takes you back to the Processes page or click Cancel to return to the Processes page without saving your changes.*

> After document sets are created, return to the Process Administration page to set the Tab Document and Tab Name.

Editing a Commerce Process

1. Modify the **Process Name** in the field provided. The process name you choose is used as the navigation link to the Commerce Manager. *This is a required field.*
2. Modify the **Description** for the process being added.

3. Choose and specify a label that you want to be displayed on the page from the **Tab Label** drop-down list.
4. Choose the number of commerce transactions you want to display on the Commerce Manager page using the **Page Length** drop-down menu. *This is a required field.*
5. Choose an **Auto Fill Option** from the drop-down menu if you want new commerce documents auto-populated with user account data or Accounts Information. *You can also opt for no auto fill by selecting None.*
6. Select one of the following actions:

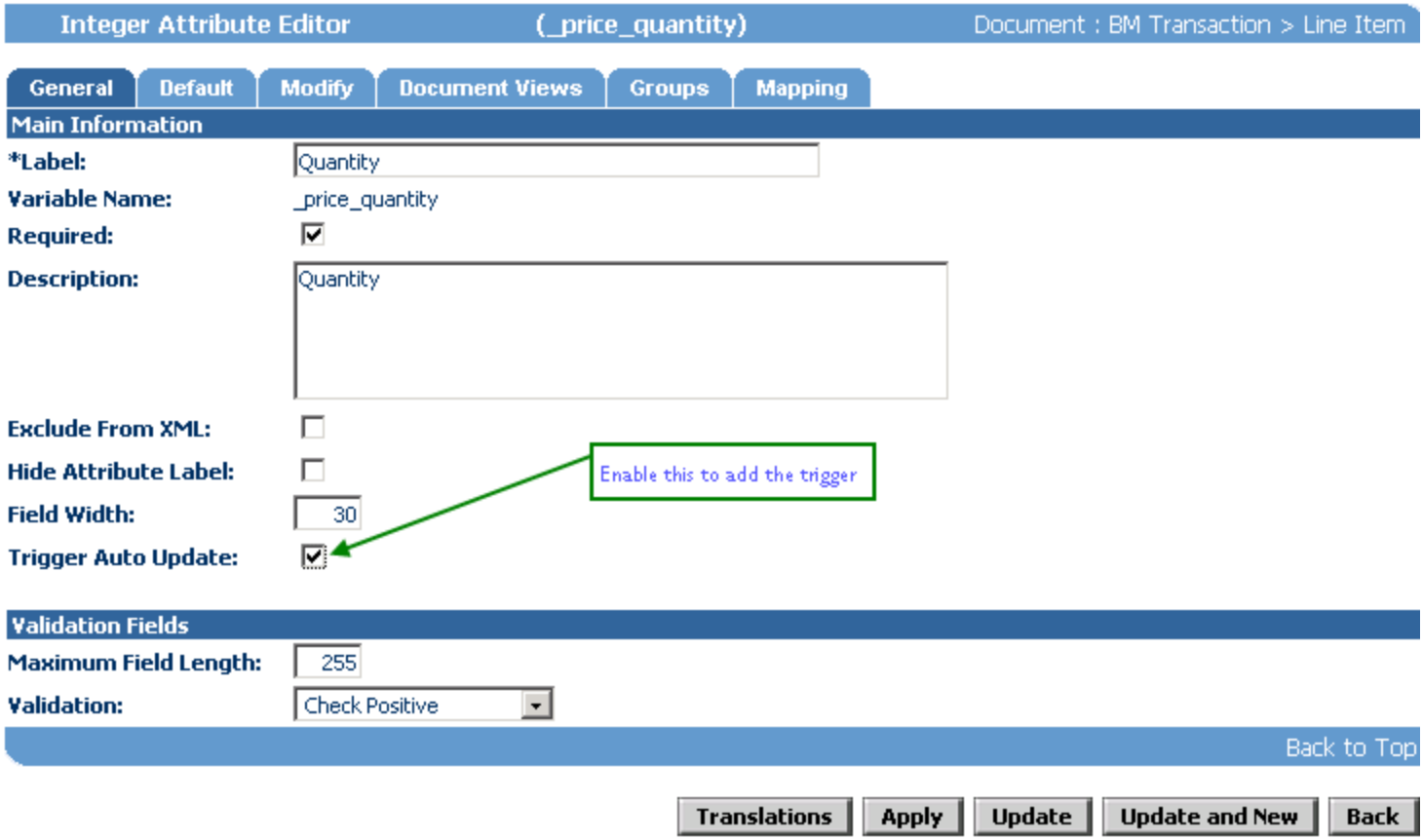
Actions	Description
Apply	This saves the changes made and refreshes the page.
Update	This updates the changes made to the database and takes you back to the Processes page.
Back	This takes you back to the Processes page.



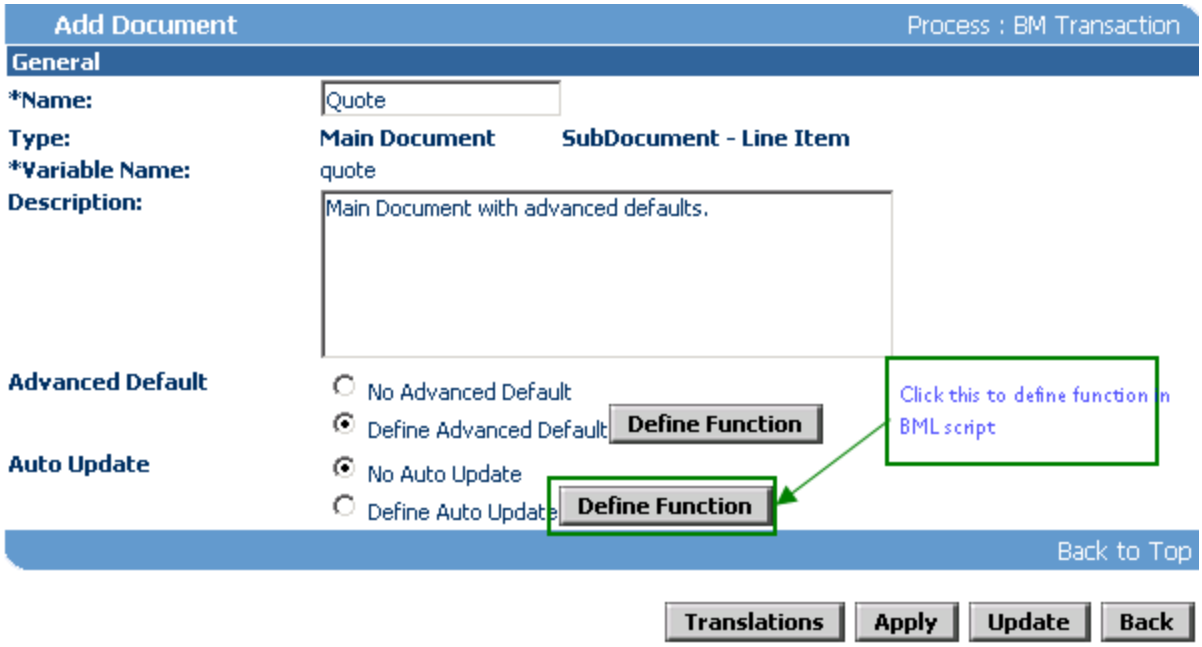
Previously created variable names cannot be edited.

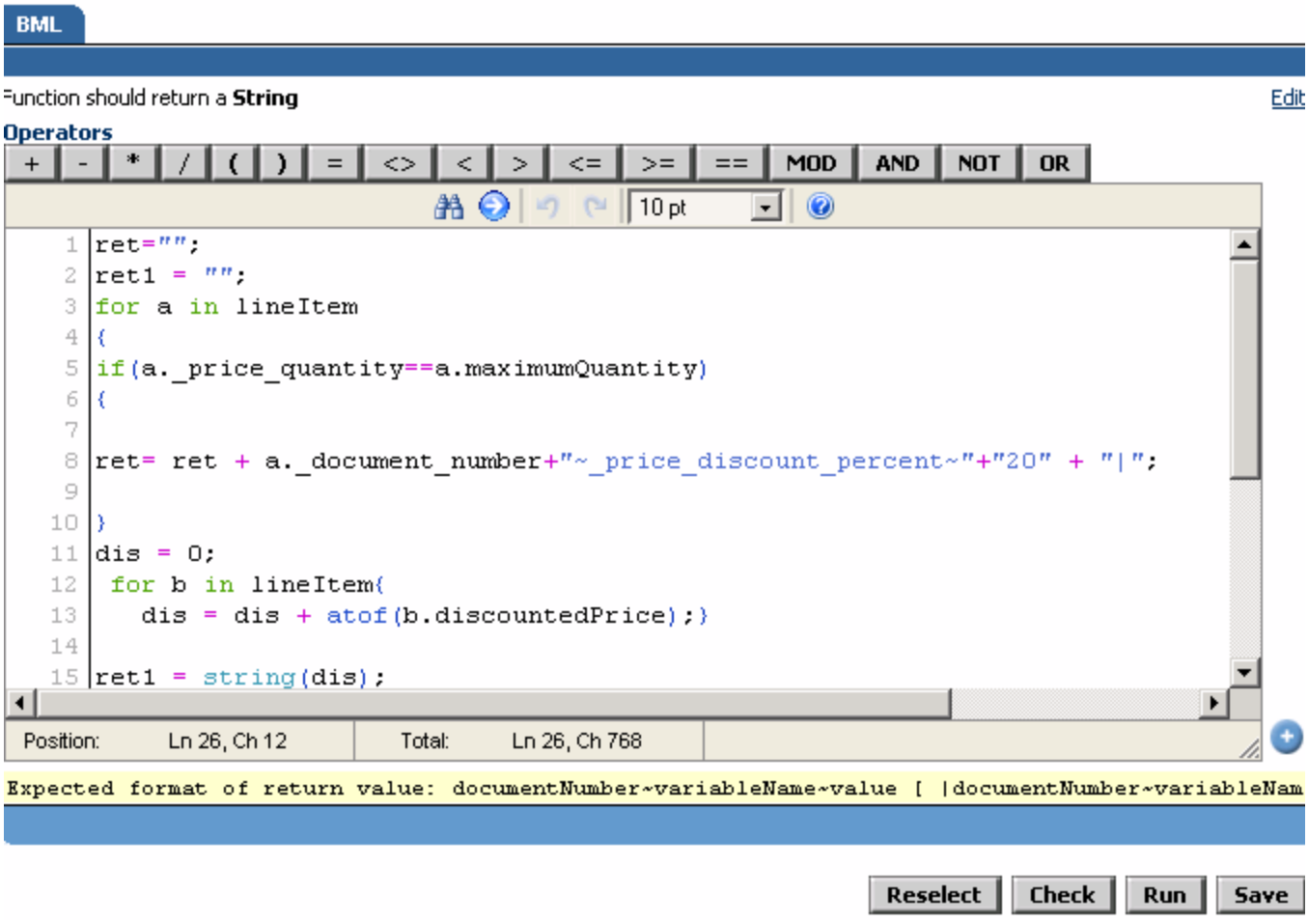
Setting up Auto-Update Fields

1. **Define Trigger Fields:**Select the **Trigger Auto-Update** option on the Attribute Editor page to make this attribute a trigger field. *When this attribute is modified, the auto-update function is run.*



2. **Define Auto-update Function:** The fullaccessuser can specify the auto-update function of the Main Document Editor.

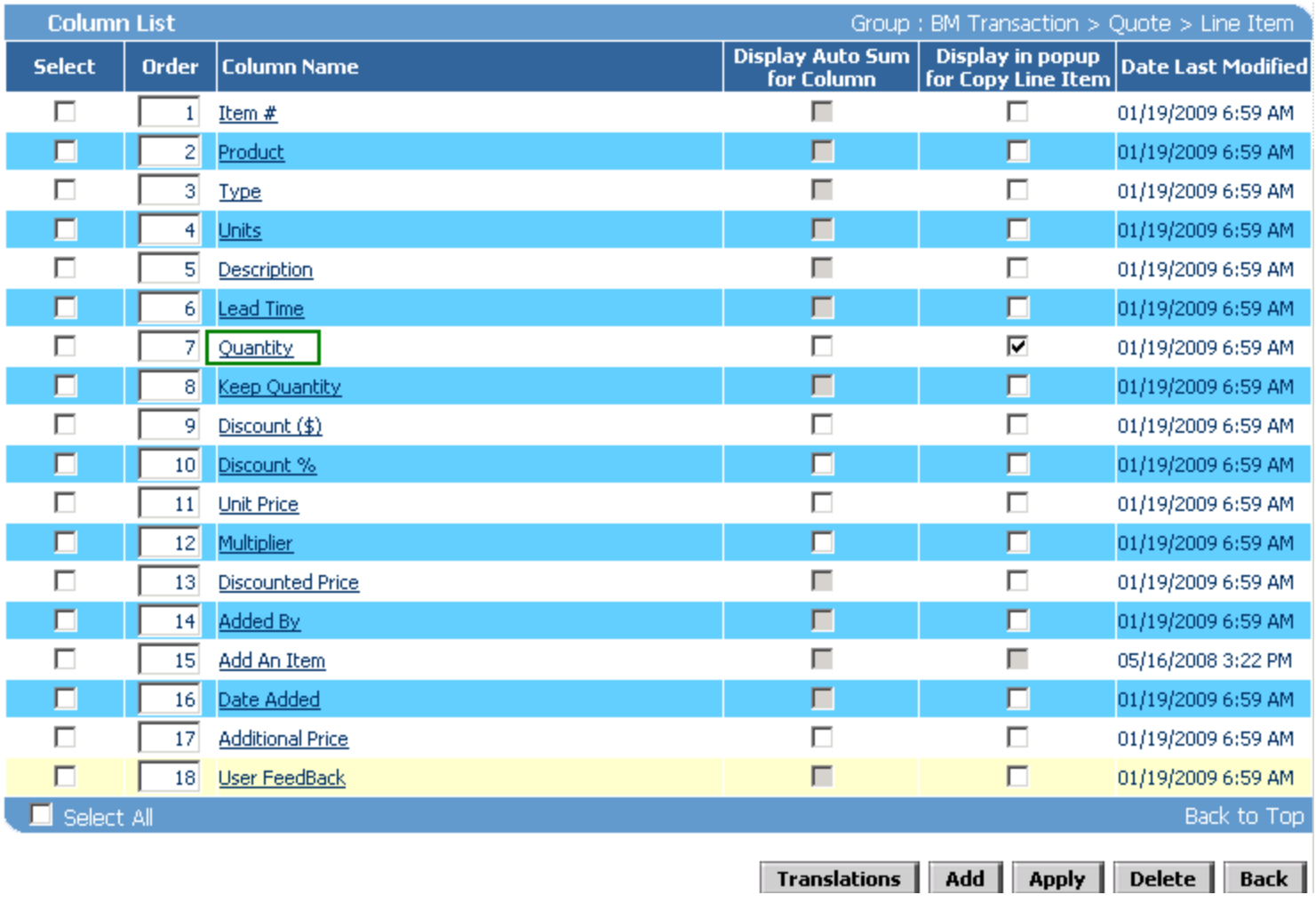

Add Document
Process : BM Transaction
General
*Name:
Quote
Type:
Main Document
SubDocument - Line Item
*Variable Name:
quote
Description:
Main Document with advanced defaults.
Advanced Default
No Advanced Default
Define Advanced Default
Define Function
Auto Update
No Auto Update
Define Auto Update
Define Function
Click this to define function in BML script
Back to Top
Translations
Apply
Update
Back


BML
Function should return a String
Edit
Operators
+ - * / ( ) = <> < > <= >= == MOD AND NOT OR
10 pt
1 ret="";
2 ret1 = "";
3 for a in lineItem
4 {
5 if(a._price_quantity==a.maximumQuantity)
6 {
7
8 ret= ret + a._document_number+"~_price_discount_percent~"+"20" + "|";
9
10 }
11 dis = 0;
12 for b in lineItem{
13 dis = dis + atof(b.discountedPrice);}
14
15 ret1 = string(dis);
Position: Ln 26, Ch 12
Total: Ln 26, Ch 768
Expected format of return value: documentNumber~variableName~value [|documentNumber~variableNam
Reselect
Check
Run
Save

The format of this rule is: `documentNumber~variableName~value [ |doc-umentNumber~variableName~value ]*`

Setting up Totals for Line Item Columns

1. Select **Display Auto Sum for Column** option on the Line Items Columns List page.



Column List — Group : BM Transaction > Quote > Line Item

Select	Order	Column Name	Display Auto Sum for Column	Display in popup for Copy Line Item	Date Last Modified
☐	1	Item #	☐	☐	01/19/2009 6:59 AM
☐	2	Product	☐	☐	01/19/2009 6:59 AM
☐	3	Type	☐	☐	01/19/2009 6:59 AM
☐	4	Units	☐	☐	01/19/2009 6:59 AM
☐	5	Description	☐	☐	01/19/2009 6:59 AM
☐	6	Lead Time	☐	☐	01/19/2009 6:59 AM
☐	7	Quantity	☐	☑	01/19/2009 6:59 AM
☐	8	Keep Quantity	☐	☐	01/19/2009 6:59 AM
☐	9	Discount ($)	☐	☐	01/19/2009 6:59 AM
☐	10	Discount %	☐	☐	01/19/2009 6:59 AM
☐	11	Unit Price	☐	☐	01/19/2009 6:59 AM
☐	12	Multiplier	☐	☐	01/19/2009 6:59 AM
☐	13	Discounted Price	☐	☐	01/19/2009 6:59 AM
☐	14	Added By	☐	☐	01/19/2009 6:59 AM
☐	15	Add An Item	☐	☐	05/16/2008 3:22 PM
☐	16	Date Added	☐	☐	01/19/2009 6:59 AM
☐	17	Additional Price	☐	☐	01/19/2009 6:59 AM
☐	18	User FeedBack	☐	☐	01/19/2009 6:59 AM

☐ Select All — Back to Top

Translations | Add | Apply | Delete | Back

This option is available for numeric or currency type attributes only.

NOTES



Processes are created in a pending state. After all components of a commerce process have been administrated, the process should be deployed.

When choosing to auto fill documents with Account data, make sure to set up mapping rules for the auto fill document action.

When performing a Mass Update, users can filter which steps the update applies to.



Commerce Processes can be added one at a time.

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""

The page length affects how fast the process page loads in the Commerce Manager. Short page lengths load quicker than longer page lengths. Document sets can be viewed by iterating through list pages.

Advanced Reports can be created with **MicroStrategy Enterprise Reporting**(third-party Reporting Software). The transaction data can be automatically mapped to the relevant fields in MicroStrategy through an integrate action. Please contact your Engagement Manager for more information.



Deployed Commerce Processes CANNOT be deleted. Use caution when deleting an undeployed commerce process because it is not possible to recover the deleted data. Upon deletion, all documents, workflow and steps contained within the process are also deleted. Upon removal of transaction, the associated Transaction Manager on the commerce side will be removed from view.

User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

RELATED TOPICS

COMMERCE PROCESS OVERVIEW

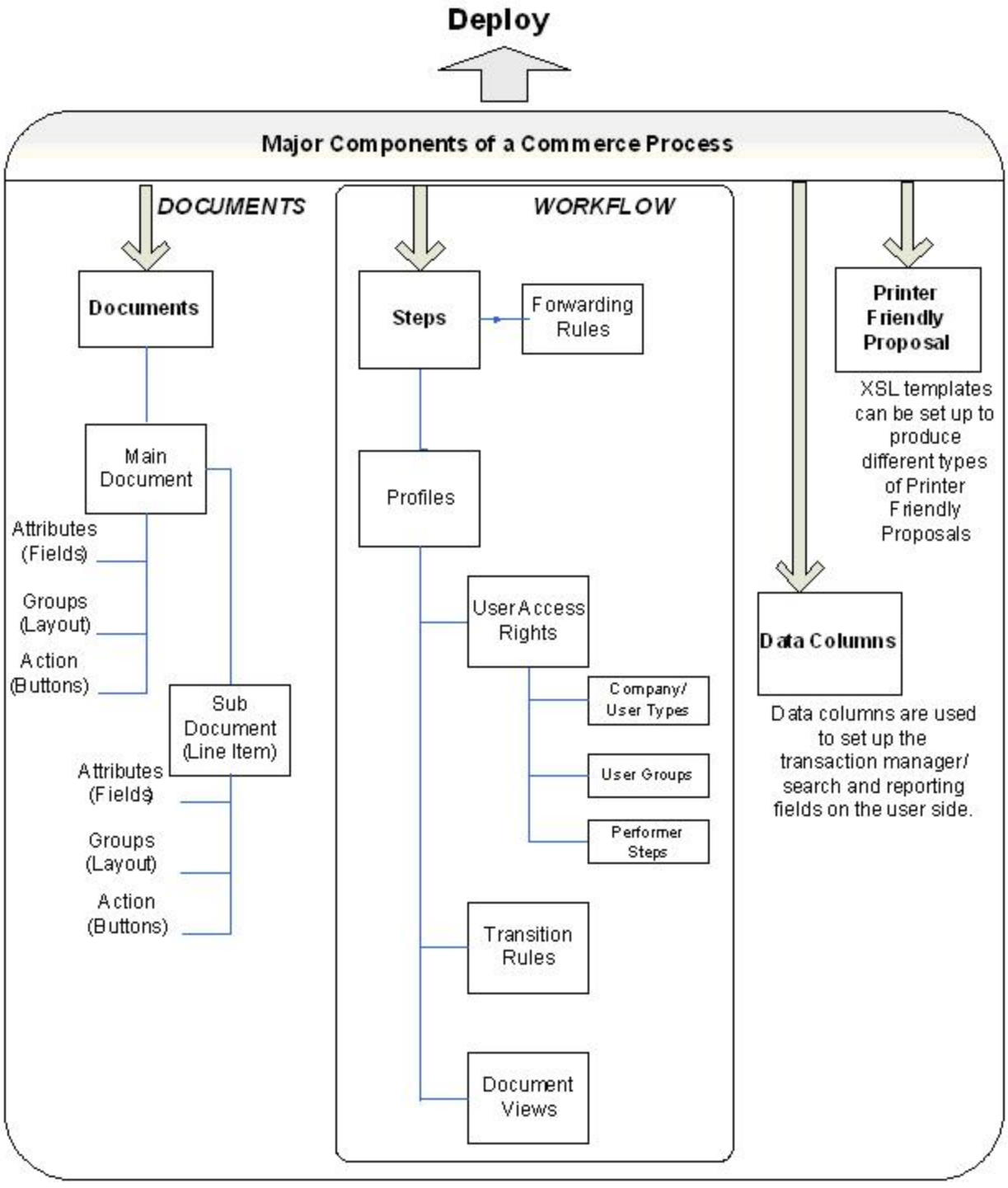
OVERVIEW

Commerce processes are used to create templates for the purchasing processes used by your company. Here, you set up your quoting, ordering, approval, and other workflow processes. By creating ordered sets of commerce document templates, associated attributes and actions, you can enable buyers and supplier agents to conduct commerce on your system. Built from the ground up, each commerce process can be customized according to how you want to structure data flow between users during transactions. You can create one process for all commerce transactions or multiple processes based on different purchase types.

- **Process Definition:** Define your commerce documents, attributes, actions and proposal views.
- **Process Invocation:** Define rules that enable the user to invoke your commerce process from various pages in the application including configuration and parts pages.

All commerce process tasks are administrated in an inactive, pending state. In order for a new process (or a change to an existing process) to be visible on the production side of the system, the process must be deployed. After a commerce process is deployed, certain commerce process administration functionality becomes hidden from view and only administrable by an implementation engineer. This happens because deployed processes contain user data once in use by partner organizations and host companies. In order to preserve user data in live commerce processes, some types of changes can only be performed on cloned processes that are then migrated into the system. At this time, the commerce migration process can only be performed by a BigMachines implementation engineer.

Commerce Process Administration Infographic



Commerce Stylesheet Editor

Stylesheets are basically CSS files that are used to set up commerce processes. This stylesheet provides an easy interface for the full-accessuser to define commerce transactions with a choice of fonts and icons. A default CSS is provided that the user can customize using the Commerce StyleSheet Editor page. **Specify Attribute Properties for:**

1. Click **Browse** to open the File Manager.
2. Scroll through the images in the File Manager. *To select an image, click the image thumbprint. This displays the selected icon next to Browse.*
3. Click **Reload** to make the change permanent. *Ensure that the image is of the size 20 pixel by 20 pixel.*
4. Click **Update** to save the changes.
5. Click **Back** to return to the previous page.

Attribute Properties	Description
Font Color	Specify the text color using the hexadecimal format. For a preview of the color, click Update<. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Font Size	Select the font-size using the drop-down list provided. You can also specify the font size in either points or pixels. Once decided, click Update to make the change permanent.
Font Style	Select the font-style using the drop-down list provided. Once decided, click Update to make the change permanent.
Font Weight	Select the font-weight using the drop-down list provided. Once decided, click Update to make the change permanent.
Font Family	Select the font-family using the drop-down list provided. Once decided upon the family, click Update to make the change permanent. *The default font is Tahoma.*

Upload icons for :

- Group Expand
- Group Collapse
- Line Item

Auto-Update Fields

Instantly update fields in the transaction main document based on trigger fields on the same document. The typical use cases might include Instant Pricing in the line item section based on changes to quantity of line items. Additionally, the admin now has the ability to display column totals for line item columns. These totals are also updated automatically.

1. This feature allows the fullaccessusers to set up calculations to update the fields on the main document when trigger fields associated with it are modified.
    - The fullaccessuser enables auto update on the transaction by checking the trigger property on attributes. *When these attributes are changed by the end user, the auto-update function is instantly run, which updates fields in the main document.*
    - The trigger attributes can be of menu, numeric, currency or boolean type and can include both main document and sub-document attributes.

- The auto-update function can update main document attributes or line item attributes that are displayed on the main document. *Only currency, integer, and text fields can be modified by the Auto-Update script.*
- The fullaccessuser must also define an auto-update function at the main document level that returns the affected attributes due to changes in the trigger fields. The return format of this rule is: `documentNumber~variableName~value [ |documentNumber~variableName~value ]*`
- Update Line Item function is not run automatically when the auto-update occurs. *The user must invoke a modify type action for the changes to be saved to the database.*

2. A feature in the line item column group allows the fullaccessuser to set up column totals.
   - When the auto-update rule fires, in addition to updating the pricing fields in the line item section (if applicable) it also updates the sum without clicking on any button.
   - A line item of numeric or currency type can be auto-summed. *This feature is not available for attributes of other types.*
3. All auto-updated fields are highlighted in sharp yellow and fade away to a dull yellow to indicate that the fields have been modified. *The dull yellow is also an indicator that the modified values have not been updated into the database until an appropriate "modify" action is clicked.*

Managing a Commerce Process

This page displays a list of the commerce processes currently available along with various Navigation options, Quick Links for the other available options, descriptions for the available processes, and the date and time the processes were Last Deployed.

Available Action	Description
Delete a Process	Select all or a required Process Name that is to be deleted using the Selection Option and click Delete to delete it.
Edit Details for Specific Processes	Click on the Process Name link to go to the Process Administration Page.
Navigate to Process-Related Sections	Select an option from the Navigation drop-down list and click List. This displays the corresponding screens for the option selected.
Quick Links	This opens a new window displaying a list of process that can be deployed together using Deploy. Click Refresh on the window to get the current data. Click Close to close it.
Reporting	This displays the Reporting Setup page for managing reports generation. This option is available if the advanced reporting module is turned on for your company.
Migration	his button is visible based on a commerce global setting under Admin Home Page > Global System Settings and Utilities > Commerce.
Add	Goes to the Process Administration page and add details for the commerce process.
Apply	Saves the changes made on the Process Administration Page.
Cancel	Returns to the Processes page without saving your changes.

Options available on the Navigation drop-down List:

- Document: Clicking List next to Documents opens the Documents List page for managing documents within the select process.

- Steps: Clicking List next to Steps displays the Steps list for managing workflows. *At least one main document must be created before Steps can be created and accessed.*
- Process Actions: Clicking List next to Process Actions displays the Process Actions List page for customizing process actions. *At least one main document must be created before Process Actions can be created and accessed.*
- Data Columns: Clicking LIst next to Data Columns displays the Mapped Data Columns page for managing data columns. *At least one main document must be created before Data Columns can be created and accessed.*
- Process Manager: Clicking LIst next to Process Manager displays the Process Manager Columns page for mapping the process columns for the Commerce Manager. *At least one main document must be created before Process Manager can be created and accessed.*
- Printer Friendly/History XSL Views
- Composite Attribute XSL Views
- Integration XSL Files
- Deployment Center



Integration is ONLY available for companies that his this module turned on. If you want to learn more about BigMachines integration module, please contact your system fullaccessuser.

ADMINISTRATION

Setting up a Commerce Process

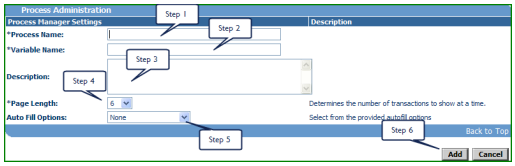
1. Create a process and save it.
2. Add main document(s) and sub document(s) to the commerce process.
3. For each document, create attributes, actions, and views.
4. Develop XSL views for print, email, and change history actions.
5. Customize process actions for the Transaction Manager.
6. Create the Transaction Manager Column layout using Data Columns.
7. Design the process workflow steps. *In each workflow step, create participant profiles to control user access. Participant profiles must be further defined using:*

- User access rights
- Transition rules
- Document views

8. Set up forwarding rules within workflow steps to automatically forward transactions to groups upon creating or saving a transaction.
9. Create timers to automatically perform document actions and auto-forwarding rules to support collaborative workflows.
10. Deploy the commerce process.

Adding a Commerce Process

1. Enter a **Process Name** in the field provided. The process name you choose is used as the navigation link to the Commerce Manager. *This is a required field.*
2. Enter a unique **Variable Name** and a **Description**. *This is a required field.*
3. Enter a description for the process being added.



4. Choose the number of commerce transactions you want to display on the Commerce Manager page using the **Page Length** drop-down menu. *This is a required field.*
5. Choose an **Auto Fill Option** from the drop-down menu if you want new commerce documents auto-populated with user account data or Accounts Information. *You can also opt for no auto fill by selecting None.*
6. Click **Add**. *This adds the new process and takes you back to the Processes page or click Cancel to return to the Processes page without saving your changes.*

> After document sets are created, return to the Process Administration page to set the Tab Document and Tab Name.

Editing a Commerce Process

1. Modify the **Process Name** in the field provided. The process name you choose is used as the navigation link to the Commerce Manager. *This is a required field.*
2. Modify the **Description** for the process being added.

3. Choose and specify a label that you want to be displayed on the page from the **Tab Label** drop-down list.
4. Choose the number of commerce transactions you want to display on the Commerce Manager page using the **Page Length** drop-down menu. *This is a required field.*
5. Choose an **Auto Fill Option** from the drop-down menu if you want new commerce documents auto-populated with user account data or Accounts Information. *You can also opt for no auto fill by selecting None.*
6. Select one of the following actions:

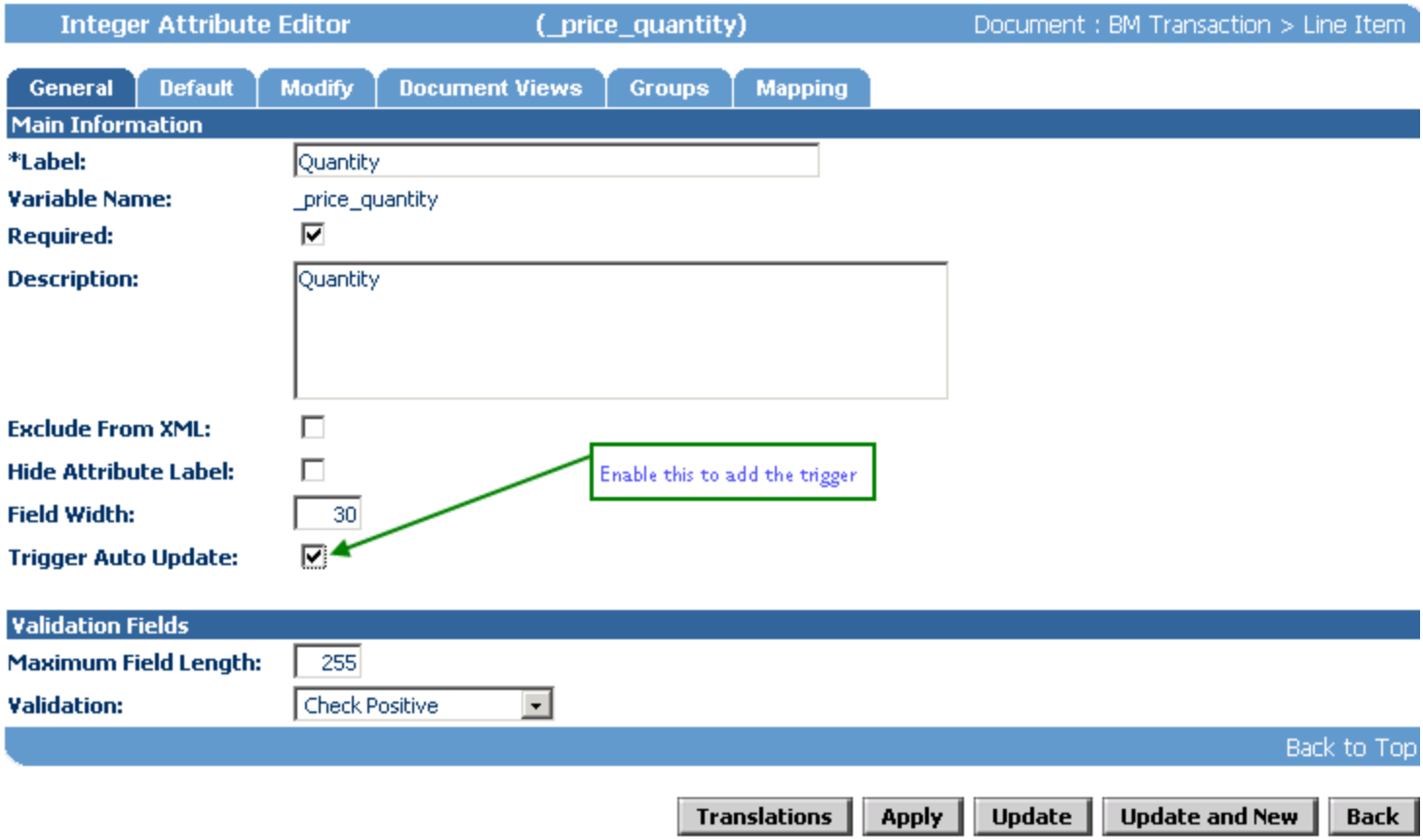
Actions	Description
Apply	This saves the changes made and refreshes the page.
Update	This updates the changes made to the database and takes you back to the Processes page.
Back	This takes you back to the Processes page.



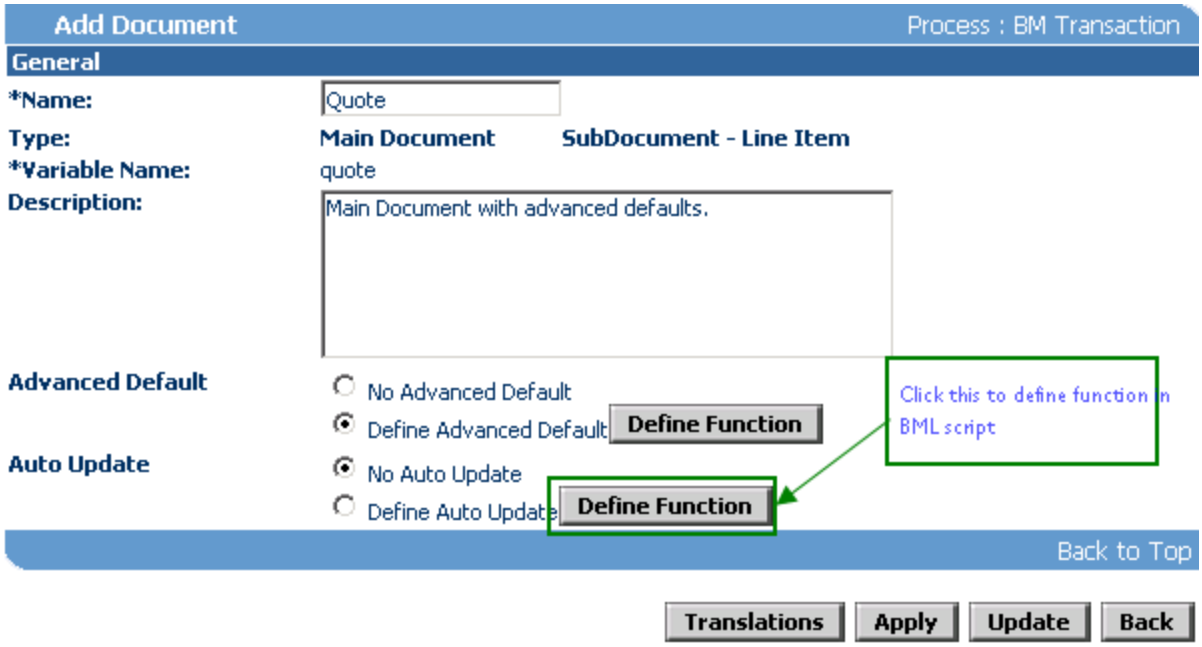
Previously created variable names cannot be edited.

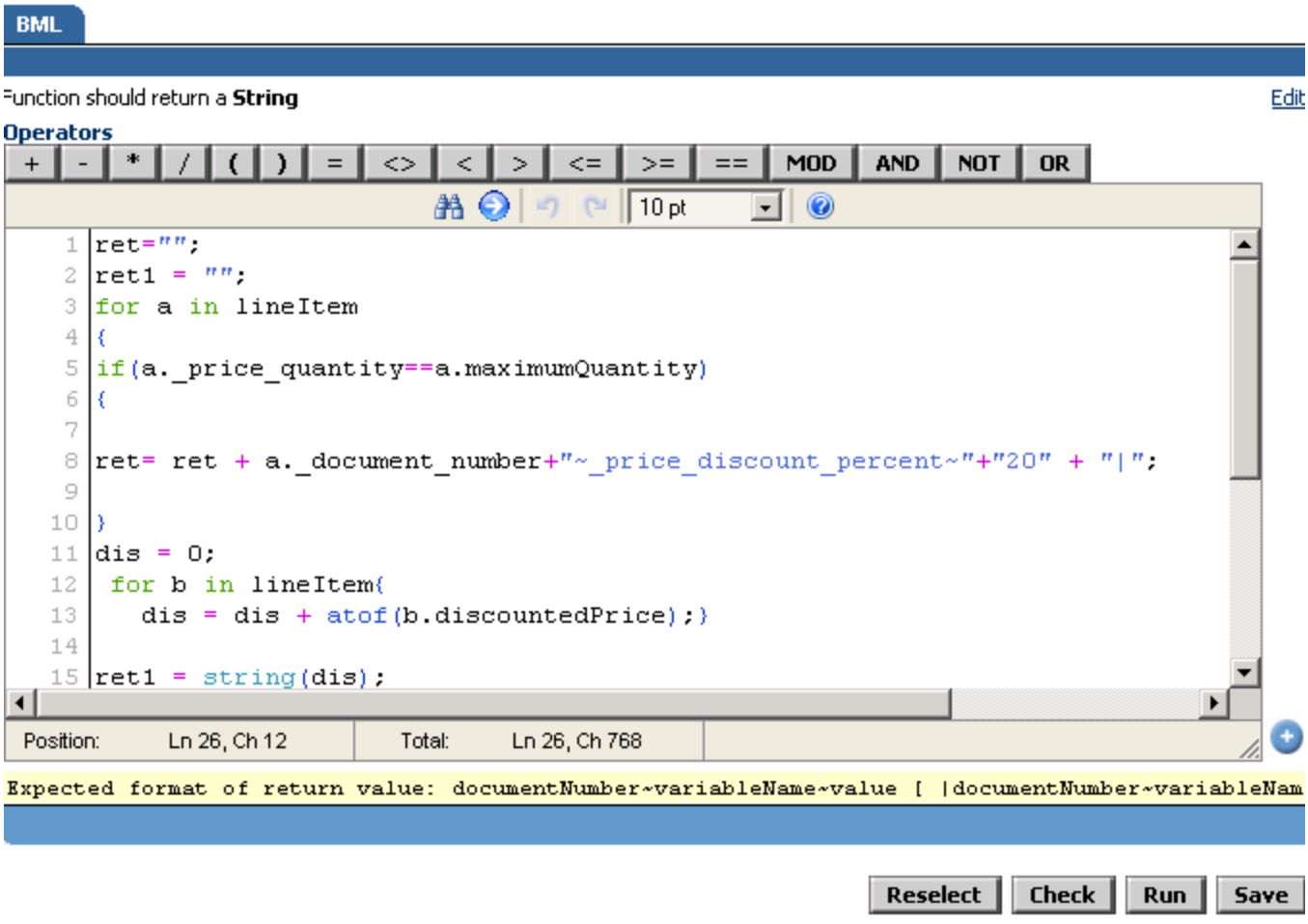
Setting up Auto-Update Fields

1. **Define Trigger Fields:**Select the **Trigger Auto-Update** option on the Attribute Editor page to make this attribute a trigger field. *When this attribute is modified, the auto-update function is run.*



2. **Define Auto-update Function:** The fullaccessuser can specify the auto-update function of the Main Document Editor.

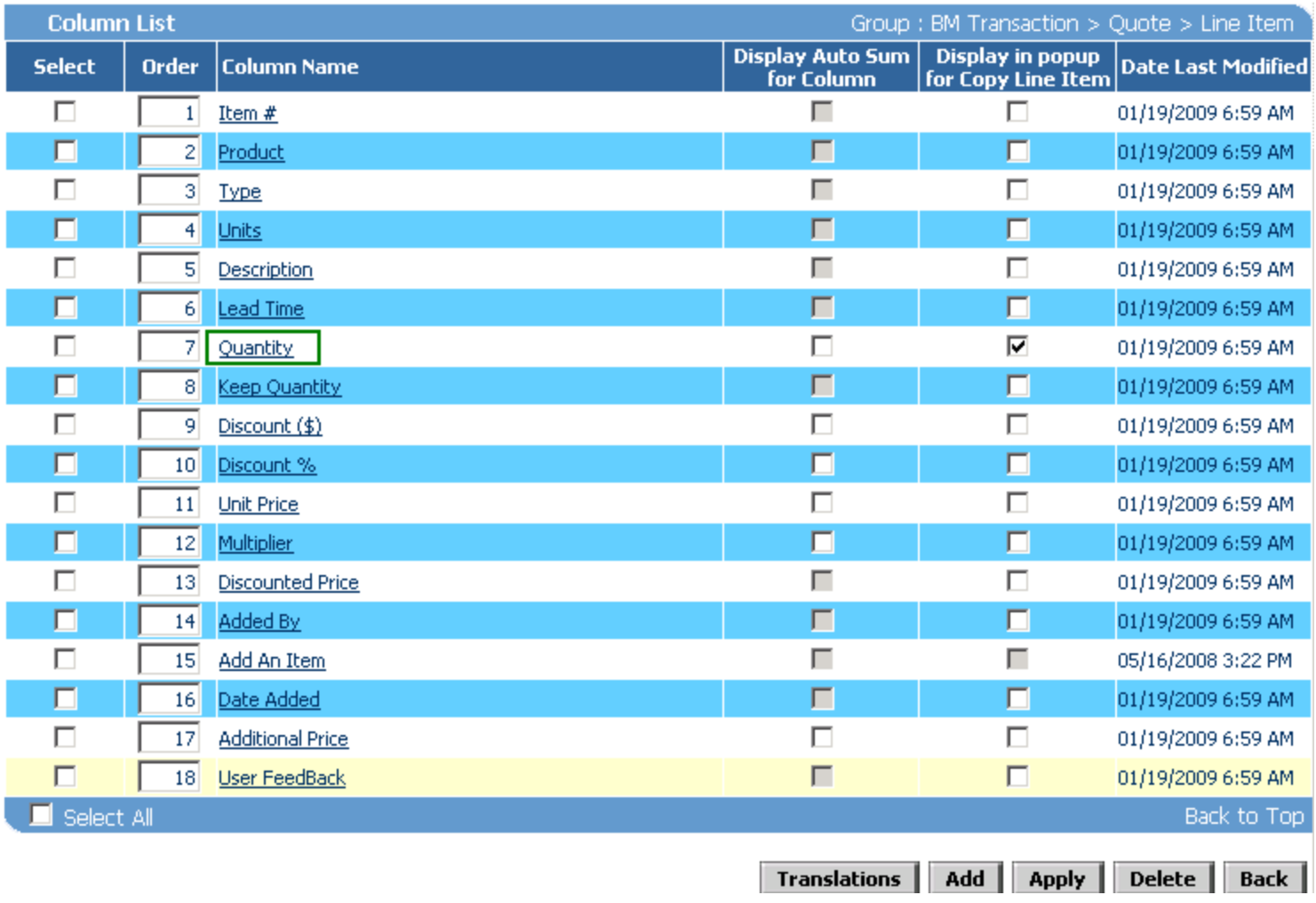

Add Document
Process : BM Transaction
General
*Name:
Quote
Type:
Main Document
SubDocument - Line Item
*Variable Name:
quote
Description:
Main Document with advanced defaults.
Advanced Default
No Advanced Default
Define Advanced Default
Define Function
Auto Update
No Auto Update
Define Auto Update
Define Function
Click this to define function in BML script
Back to Top
Translations
Apply
Update
Back


BML
Function should return a String
Edit
Operators
+ - * / ( ) = <> < > <= >= == MOD AND NOT OR
10 pt
1 ret="";
2 ret1 = "";
3 for a in lineItem
4 {
5 if(a._price_quantity==a.maximumQuantity)
6 {
7
8 ret= ret + a._document_number+"~_price_discount_percent~"+"20" + "|";
9
10 }
11 dis = 0;
12 for b in lineItem{
13 dis = dis + atof(b.discountedPrice);}
14
15 ret1 = string(dis);
Position: Ln 26, Ch 12
Total: Ln 26, Ch 768
Expected format of return value: documentNumber~variableName~value [|documentNumber~variableNam
Reselect
Check
Run
Save

The format of this rule is: `documentNumber~variableName~value [ |doc-umentNumber~variableName~value ]*`

Setting up Totals for Line Item Columns

1. Select **Display Auto Sum for Column** option on the Line Items Columns List page.



Column List — Group : BM Transaction > Quote > Line Item

Select	Order	Column Name	Display Auto Sum for Column	Display in popup for Copy Line Item	Date Last Modified
☐	1	Item #	☐	☐	01/19/2009 6:59 AM
☐	2	Product	☐	☐	01/19/2009 6:59 AM
☐	3	Type	☐	☐	01/19/2009 6:59 AM
☐	4	Units	☐	☐	01/19/2009 6:59 AM
☐	5	Description	☐	☐	01/19/2009 6:59 AM
☐	6	Lead Time	☐	☐	01/19/2009 6:59 AM
☐	7	Quantity	☐	☑	01/19/2009 6:59 AM
☐	8	Keep Quantity	☐	☐	01/19/2009 6:59 AM
☐	9	Discount ($)	☐	☐	01/19/2009 6:59 AM
☐	10	Discount %	☐	☐	01/19/2009 6:59 AM
☐	11	Unit Price	☐	☐	01/19/2009 6:59 AM
☐	12	Multiplier	☐	☐	01/19/2009 6:59 AM
☐	13	Discounted Price	☐	☐	01/19/2009 6:59 AM
☐	14	Added By	☐	☐	01/19/2009 6:59 AM
☐	15	Add An Item	☐	☐	05/16/2008 3:22 PM
☐	16	Date Added	☐	☐	01/19/2009 6:59 AM
☐	17	Additional Price	☐	☐	01/19/2009 6:59 AM
☐	18	User FeedBack	☐	☐	01/19/2009 6:59 AM

☐ Select All — Back to Top

Translations | Add | Apply | Delete | Back

This option is available for numeric or currency type attributes only.

NOTES



Processes are created in a pending state. After all components of a commerce process have been administrated, the process should be deployed.

When choosing to auto fill documents with Account data, make sure to set up mapping rules for the auto fill document action.

When performing a Mass Update, users can filter which steps the update applies to.



Commerce Processes can be added one at a time.

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""

The page length affects how fast the process page loads in the Commerce Manager. Short page lengths load quicker than longer page lengths. Document sets can be viewed by iterating through list pages.

Advanced Reports can be created with **MicroStrategy Enterprise Reporting**(third-party Reporting Software). The transaction data can be automatically mapped to the relevant fields in MicroStrategy through an integrate action. Please contact your Engagement Manager for more information.



Deployed Commerce Processes CANNOT be deleted. Use caution when deleting an undeployed commerce process because it is not possible to recover the deleted data. Upon deletion, all documents, workflow and steps contained within the process are also deleted. Upon removal of transaction, the associated Transaction Manager on the commerce side will be removed from view.

User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

RELATED TOPICS

COMMERCE PROCESS SETTINGS

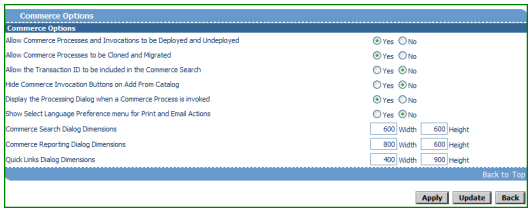
OVERVIEW

Use this page to customize commerce process settings. Options are available for allowing commerce processes to be deployed, cloned, and migrated. You can also set the transaction id to be displayed on the search popup. There are additional options for hiding commerce invocation buttons on 'Add from Catalog' as well as ability to specify popup dimensions for search dialog and quick links.

ADMINISTRATION

Setting Commerce Options

1. You can customize the look and feel of your commerce process settings by specifying the following options:
    - Allow Commerce Processes and Invocations to be Deployed and Undeployed: Select **Yes** to accept and **No** to reject.
    - Allow Commerce Processes to be Cloned and Migrated: Select **Yes** to accept and **No** to reject.
    - Allow the Transaction ID to be included in the Commerce Search: Select **Yes** to accept and **No** to reject.
    - Hide Commerce Invocation Buttons on Add From Catalog: Select **Yes** to accept and **No** to reject.
    - Display the Processing Dialog when a Commerce Process is invoked: Select **Yes** to accept and **No** to reject.
    - Allow users to print/email transactions in all supported languages by selecting **Yes** for the Show Select Language Preference menu for Print and Email Actions option.
    - Commerce Search Dialog Dimensions: Enter the **Width** and **Height** for the dialog dimensions.
    - Quick Links Dialog Dimensions: Enter the **Width** and **Height** for the dialog dimensions.
2. Click:
    - **Apply** to save your changes and remain on the current page.
    - **Update** to save your changes and return to the Admin home page.
    - **Back** to return to the Admin home page.



RELATED TOPICS

COMMERCE RULES

OVERVIEW

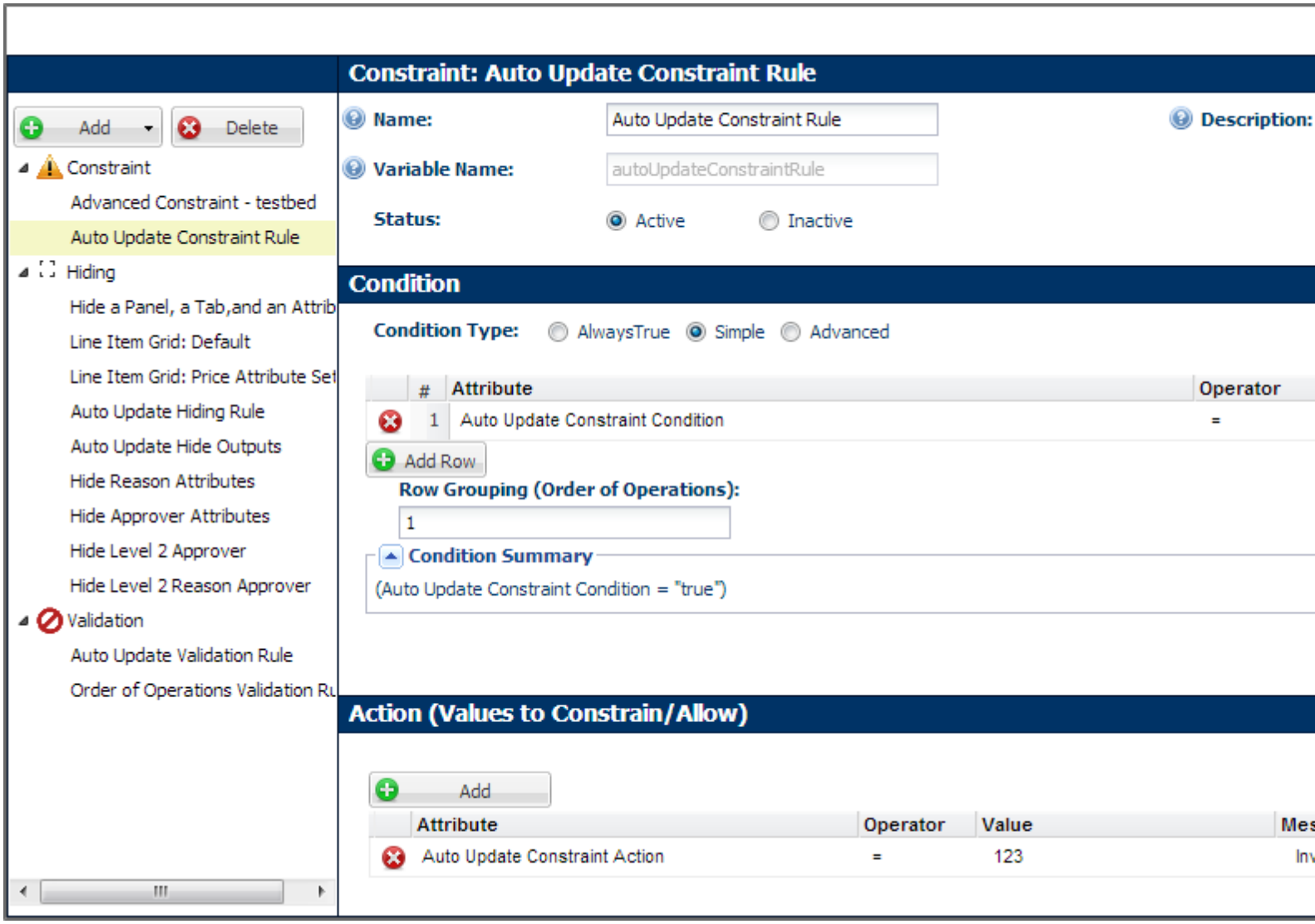
BigMachines provides the ability to create commerce rules that are similar to the configuration rules. Commerce rules allow you to do things like limit the values in one field based on the selections in another and hide attributes based on other selections. They can be created at the main-document and sub-document levels. Rules that are created on the main-document or sub-document levels can affect attributes on either level. **Example:** you can write a constraint rule where the condition attribute is on the main-document and the action attribute is on the sub-document. Users have the ability to create, edit and view all commerce rules from a central location.

Available Commerce Rules

Rule Type	Description	Example
Hiding	Hides attributes when a pre-defined condition is met. These rules are triggered based off an attribute change (AJAX).	If the Line Item Grid contains a line with Part Number 'S2000', then hide that line's Net Ea input attribute and Net Ea Override Attribute.
Constraints	Determines attribute values to constrain or allow OR fields that can't be populated in commerce. These rules are triggered based off an attribute change (AJAX).	If the Bill To Company Name = "ACME", constrain all non-zero values in the Discount column on the Line Item Grid. Include message "No discounts allowed for ACME'.
Validations	Validates attribute or field values. Rules are linked to an action and only run when a specific action is clicked by the user.	The user must populate Quote Description [quoteDescription_quote] with something before they click "Submit". If the validation fails, show the user a message next to the attribute: 'Quote Description must be populated prior to quote submittal'.

Commerce Rule Editor

Commerce rules are all created, edited and viewed from a central location, similar to the data table repository. This allows you to create all of your rules in one place.



Main-Document Rules

Rules created at the main-document level can affect both main-document and sub-document attributes. *When a rule is created at this level but run on a sub-document attribute within the line item grid, the rule action will be performed on the entire column within the line item grid.* **Example:** if you would like to hide "Discount" for a particular part number. If the rule condition is met anywhere in the line item grid, the entire column will be hidden for all line items.

- Main-Document rules using sub-document attributes in the condition will fire if any life item in the transaction meets the condition.
- Main document rules that use line level attributes in the action act on line level attributes will act on all line items in the transaction.

- Main document hiding rules can be used to hide a column in the line item grid by hiding the line level attribute mapped to that column. Similarly, main document validation and constraint rules can be used to constrain all values of a column in the line item grid by constraining the attribute mapped to the column.



Rules created at this level have access to all system, main document and line level attributes in the condition. These rules can only hide, constrain or validate line level attributes.

Sub-Document Rules

Rules created at the sub-document level can only affect sub-document attributes. This functionality allows for rules to evaluate each line within the line item grid individually. **Example:** if you wanted to hide "Discount" for a particular part number, the rule would evaluate each line item and if the condition is met, only that line item is affected. In this case, the input field for Discount would be removed for that particular line item.

- Users can observe line level rules fire on the main document if the action attribute is mapped to the line item grid.
- Line level rules impact individual line items.
    - *If the condition of a line level rule is based on a line level attribute (or a combination of main document and line level attributes), then only the value of the attribute for the current line item is considered when the rule is evaluated.*
    - *If the condition of a line level rule is Always True or only has main document attributes as input, then all line items will be impacted.*
- Line level validation rules can be run with both main document and line level actions.

Rule Behavior

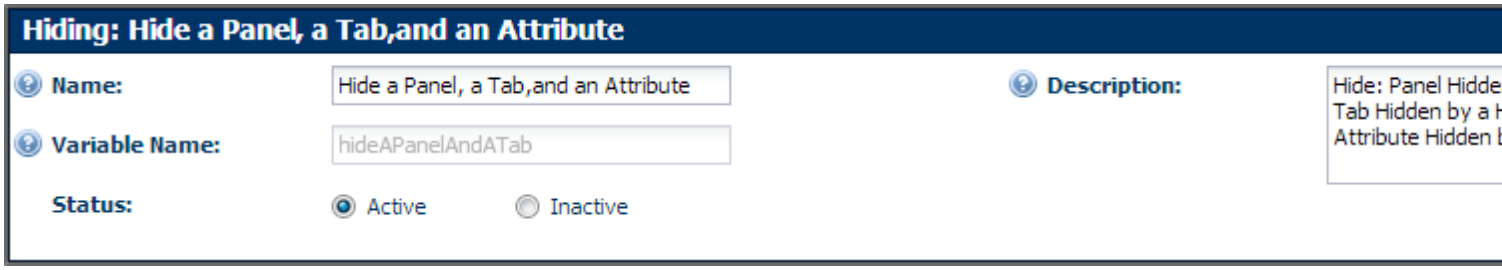
Similar to configuration rules, a commerce rule will fire when the condition evaluates to true. The condition of an AJAX enabled commerce rule is re-evaluated when any of its input attributes is changed either by the user or by the auto-update function. Input attributes include condition and action attributes and any attributes used as inputs to the condition or action advanced function. Commerce rules also function when the user interacts with BigMachines via SOAP. They even fire when the condition or action attributes are not mapped to the layout, or may not be visible.

New Validation rules are evaluated and executed after the action is performed. New Validation rules associated with Update Line Item fire when the action is called implicitly or when another modify-type action is performed. Simple validations, like required, range check, etc., will fire automatically for attributes that are associated with hiding or constraint rules. For attributes not associated with rules, these validations fire when an action is performed. For SOAP, rule information is embedded in `GetTransaction` and `CreateTransaction` responses and validation message in `UpdateTransaction` response. Advanced validations are being deprecated for new implementations. Use standard validations instead.

Commerce Rule Elements: A commerce rule is comprised of a condition and an action. The condition is always evaluated first. If the condition is evaluated to be true (is met), then the rule is fired (action will be performed). *Each of the rules start with a basic template of three sections:*

- **Properties**

This section is where you will enter the basic details about the rule (for example, the name, status and where it will be applied).



Label	Description
Name	Enter the name of the rule in this section. *This is a required field.*
Variable Name	Enter the variable name, but remember that this cannot be changed. *This is a required field.*
Description	Enter a description, if desired. Only viewable by fullaccessusers, use this field to describe how the rule functions so that other fullaccessusers can easily understand the purpose of the rule.
Status	*Determine the status of the rule:* **Active**- the rule is fired when the condition is met. **Internal**- the rule is active only for BigMachines FullAccessUser. The rule is inactive for all others. **Inactive**- the rule is not fired even if the condition is met. Inactive rules are displayed in italics in the rules list in the left panel.

- ## Condition

The condition section allows you to select the logic that specifies when the rule should run.



Condition Type	Description
Always True	The rule will fire automatically, every time, because a specific condition does not need to be met.
Simple	Here you can use the rule editor to define condition attribute values. You can choose from System, Main-Document and/or Sub-Document attributes.
Advanced	Here you can define advanced conditions using BML.

- ## Action

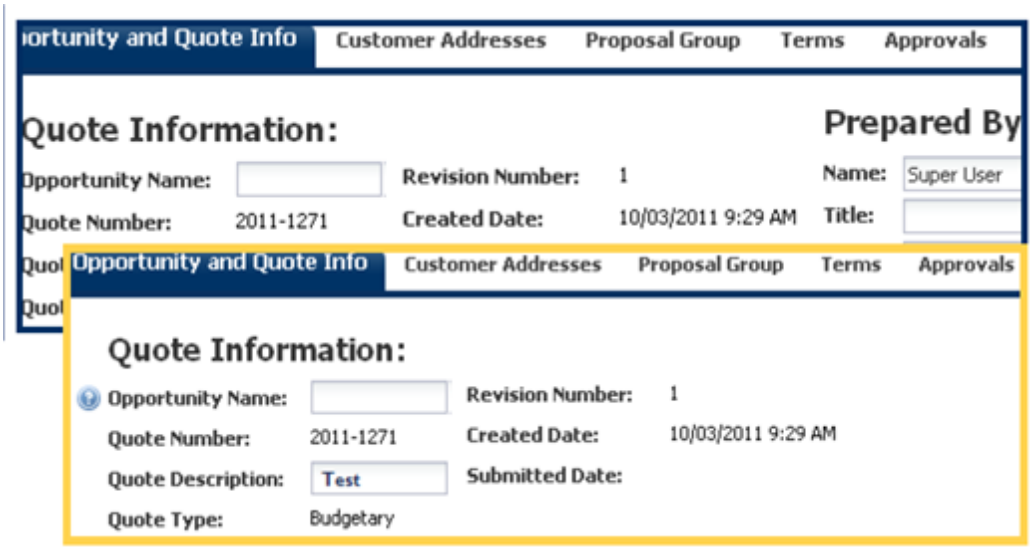
The action section is where you select WHAT happens if the condition is met. All rule types can be written using the simple action editor.



ADMINISTRATION

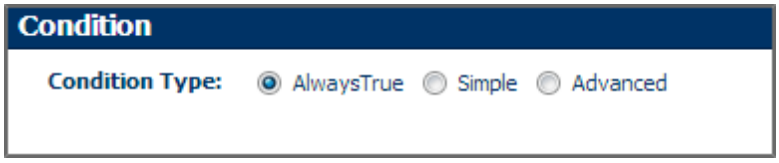
Hiding Rules

Similar to hiding rules in configuration, hiding rules in commerce are created to hide attributes from the buyer when certain conditions are met. *Commerce hiding rules are run using AJAX, meaning that an action doesn't need to be performed in order for them to run.* A hiding rule has a condition and an action. When the condition is met, the rule will hide the action attribute(s). Hiding rules can be written and/or run on attributes in the main-document or sub-document. Different from constraints, hiding rules written on the sub-document level can only hide sub-document attributes.



Adding a Hiding Rule

1. Navigate from: Process Definition > Select Documents for your Commerce Process > **Rules** *(choose Main-Document or Sub-Document)*
2. Click **Add** to create a new rule and select Hiding.
3. Enter the basic **Properties** of the rule.
4. Create your **Conditions**. *The value is case-sensitive.*



5. Click **Add** under the Action (Attributes to Hide) section.



6. Select **Main Document** or **Sub Document** attributes to hide. *If you are writing a rule on the sub-document, you will only be able to hide sub-document attributes.*

7. Click **Save** . *You will now see your rule appear in the left panel under Hiding.*

Hiding Rules on Sub-Document Level

The image below shows you how a hiding rule on the sub-document would look like to the user. In this case, a rule was written so that when the part number was S2000, the Net Ea Write-In and Net Ea Override attributes where affected. Since the entire column should not be removed, as this rule evaluates all lines and applies it only when the condition is met, the application will remove/hide the input fields for the affected attributes.



	#	Qty	Item	Item Descr	Price	$ or %	Disc	Net Each Write-in	Net Ea Override
	1	1	HW	Hardware 00006	$600.00		0.0	$0.00	
	2	1	S2000	It's a brand new car!	$1.00		0.0		

You can control whether or not a dialog loading box appears for the constraint and hiding rules by making a settings change within **Commerce Settings**. For the option 'Number of Milliseconds to Wait Before Showing the Loading Dialog for AJAX Rules', enter:

- **Positive number:** Determines how many milliseconds the dialog will appear.
- **Negative number:** Disables the processing dialog for AJAX rules.
- **Zero (0):** This will happen instantaneously.

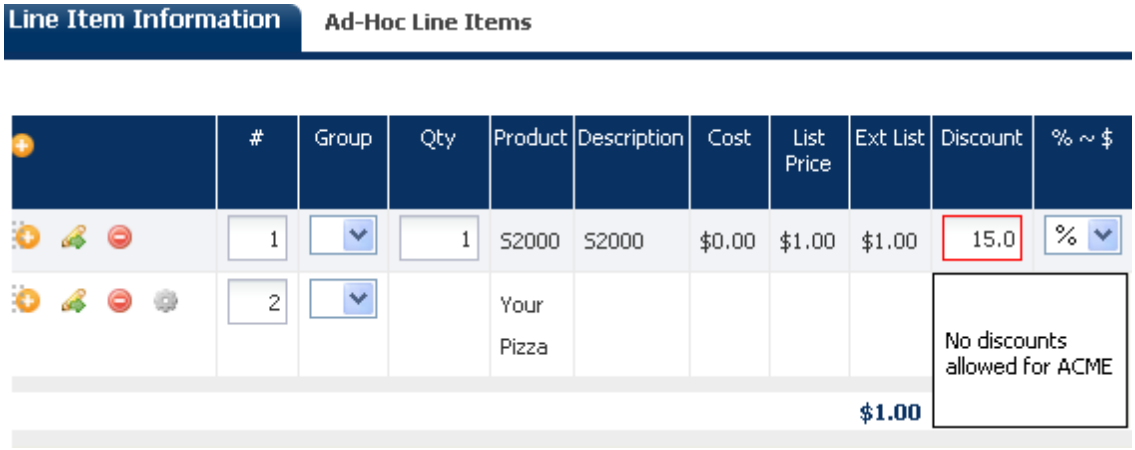
In order to remove an entire column from the line item grid, you'll have to write a hiding rule on the main-document level that affects a sub-document attribute.

Hiding rules are performed with AJAX functionality, while validation rules are tied to actions, meaning a user must click an action button for validation rules to run.

Because of AJAX functionality, an action doesn't need to be performed in order for hiding rules to run. In addition, the rule will run without refreshing the page.

Constraint Rules

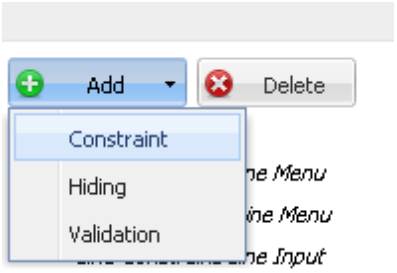
Similar to constraint rules in configuration, constraint rules in commerce are set up to warn the buyer when certain attribute values are not allowed. While a constraint is active, the system doesn't allow the user to advance within the commerce process (i.e., saving a quote). Once the user has alleviated the constraint, he is allowed to use any displayed commerce actions to proceed through the quoting process. A constraint rule has a condition and an action. When the condition and action are both true, then a constraint is fired, which shows up as a message to the user. *Constraints can be written and/or run on attributes in the main-document or sub-document.* **Example:** an attribute value on the main-document could constrain a value in the line item grid and vice-versa.



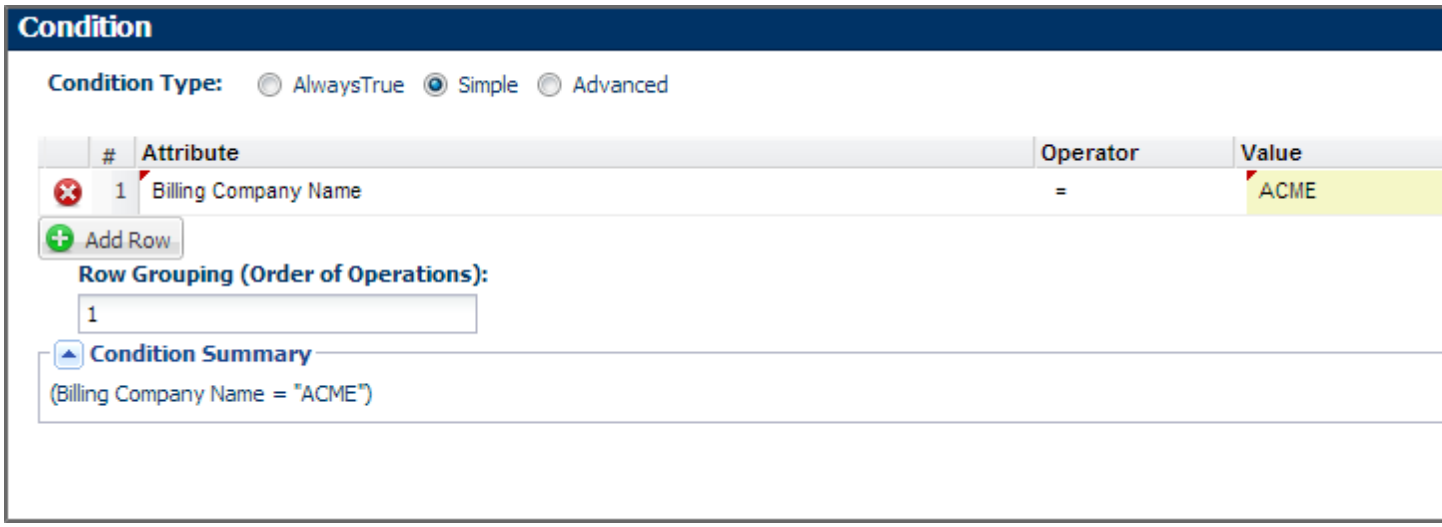
Adding a Constraint Rule

Example Use Case: If the Bill To Company Name is "ACME", then we want to constraint all non-zero values in the Discount column in the line item grid. An error message will appear that reads: "No Discounts allowed for ACME". *The result will display the same as the image above.* We are using a main-document attribute to constrain the value of a line item attribute.

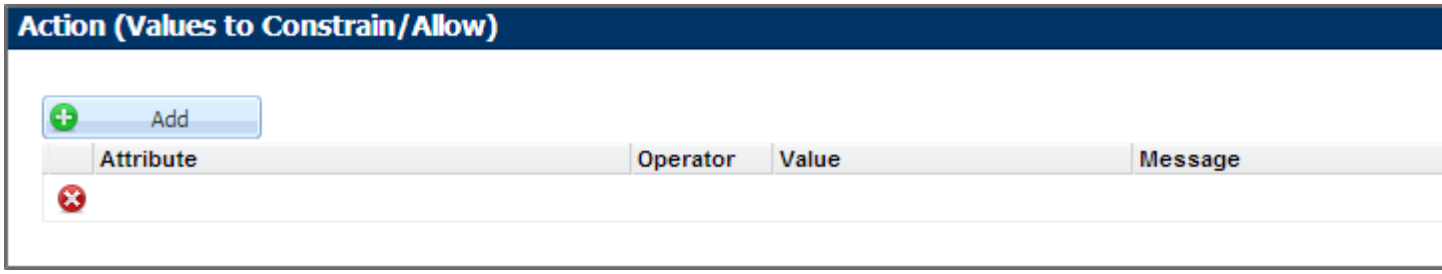
1. Navigate from: Process Definition > Select Documents for your Commerce Process > **Rules** *(choose Main-Document or Sub-Document)*



2. Click **Add** to create a new rule and select **Constraint**.
3. Enter the basic **Properties** of the rule.
4. Create your **Conditions**. *This rule should only run when a user enters ACME into the Bill To Company Name.* Make sure to select the Main Document Folder and search for the attribute Company Name `[_billTo_Company_Name]`.



5. Select the **Action Type**. *In this case, you'll select Simple.*
6. Click **Add**.
7. Select a Main Document or Sub Document attribute and its value to constrain or allow. Available attributes: integer, float, currency, text, boolean, menu, and date. *In this case, the attribute to select is Discount* `[discount_line]`.
8. Enter a constraint message to display to the user. *Use the drop-down menu to select where you'd like to display the message.*



9. Click **Save**. *You will now see your rule appear in the left panel under Constraints.*

Behavior: The time component of date attributes are not supported in commerce rules. Currency attributes with a default of 0.0 are not constrained in a new transaction even if the value is constrained by a rule. You can set up a validation rule to ensure that the transaction cannot be saved with this value. For an integer attribute constraining 0 will not constrain the blank value. If you constrain a currency attribute using a simple constraint rule, then the exchange rate will be taken into account when the attribute is constrained. **Example:** in a USD base currency site if the constrained value of a currency attribute is 1 and the exchange rate for Euro is 2, then for the Euro user the constrained value will be 2 while for the dollar user, the constrained value will be 1. However, if you use an advanced constraint rule, the exchange rate will not be taken into account. In this case, if the function returns 1 as the constrained value, then the value 1 will be constrained for both the dollar and the Euro user.

You can control whether or not a dialog loading box appears for the constraint and hiding rules by making a settings change within **Commerce Settings**. For the option 'Number of Milliseconds to Wait Before Showing the Loading Dialog for AJAX Rules', enter:

- **Positive number:** Determines how many milliseconds the dialog will appear.
- **Negative number:** Disables the processing dialog for AJAX rules.
- **Zero (0):** This will happen instantaneously.



Use a constraint rule if you'd like the user to correct a constrained value prior to clicking any commerce action.

Actions that are "Save Without Validating" or "Modify Without Saving or Validating" type can be performed even if there is a constraint on the page.

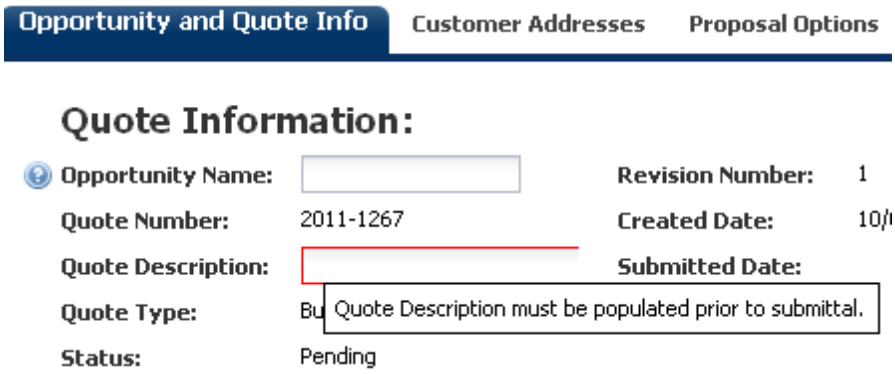


Constraint rules are performed with AJAX functionality, while validation rules are tied to actions, meaning a user must click an action button for validation rules to run. AJAX allows the rule to run without requiring the user to refresh the page.

Reverse constraints are allowed for menu-type attributes when you can specify the allowed values instead of the constrained values.

Validation Rules

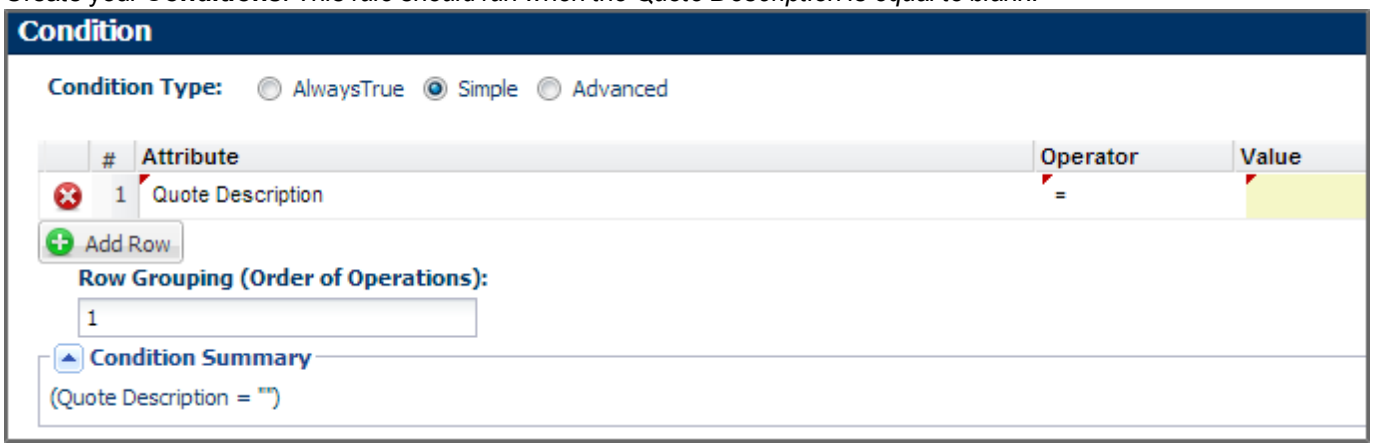
Validation rules will validate the attribute values in the context of the current document, and its associated parent/child, when the user clicks an action button. **Example:** you can write a rule to validate that the user has entered something into the Quote Description. A validation rule has a condition and an action. When the condition is met, the validation runs on the action attribute value(s). Any constrained values will show up as a message to the user. *Validations can be written and/or run on attributes in the main-document or sub-document.* **Example:** an attribute value on the main-document could constrain a value in the line item grid and vice-versa.



Adding a Validation Rule

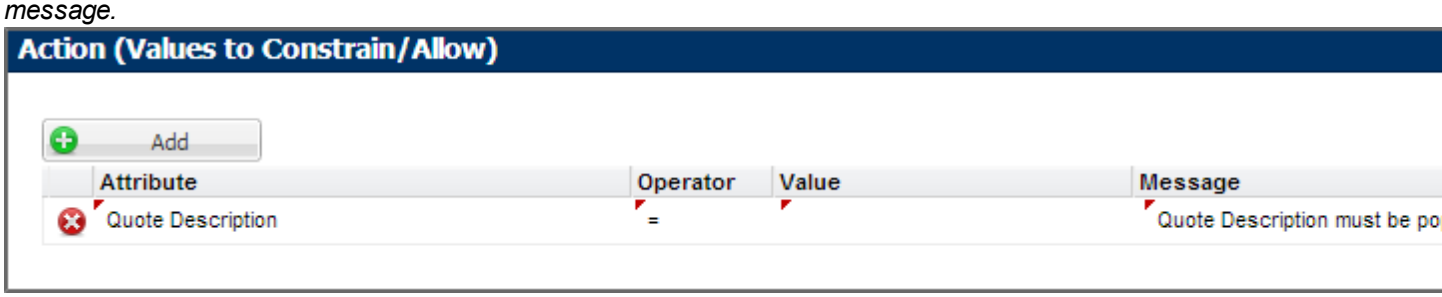
Write a validation to ensure that the user populates Quote Description with something before they click Submit. If the validation fails, show the user a message next to the attribute that reads, “Quote Description must be populated prior to quote submittal”.

1. Navigate from: Process Definition > Select Documents for your Commerce Process > **Rules** *(choose Main-Document or Sub-Document)*
2. Click **Add** to create a new rule and select **Validation**.
3. Enter the basic Properties of the rule.
4. Select the **Available Actions**A for which to add this validation to. *For this use case, we’d like the validation to only run when the user clicks Submit.*
5. Create your **Conditions**. *This rule should run when the Quote Description is equal to blank.*



6. Select the **Action Type**. *In this case, you’ll select Simple.*
7. Click **Add**.
8. Select a Main Document or Sub Document attribute and its value to constrain. *In this case, the attribute to select is Quote Description which will be equal to blank.*

9. Enter the constraint message to display to the user. *Use the drop-down menu to select where you'd like to display the message.*



10. Click **Save**.

Use a validation rule if you'd like to allow the user to perform an action (i.e. Save) on a quote with that has a constrained value.

Advanced validations will no longer be available on commerce actions. All validations will be written from the commerce rule editor page. *However, if you are upgrading from an earlier version of BigMachines, your current advanced validations will remain in place, until you migrate to the new commerce validation rules.*

NOTES

Use validation rules instead of advanced validations in actions. All validations will be written from the commerce rule editor page. *However, if you are upgrading from an earlier version of BigMachines, your current advanced validations will remain in place, until you migrate to the new commerce validation rules.*

Test performance early and often – AJAX enabled constraint and hiding rules might make the transaction page slow depending on the number of attributes, rules, use of advanced function etc. Performance tests are a must!



The order of operations will be modified and perform as follows: Simple validations will be evaluated first, followed by simple modifications. Then advanced modifications and auto-updates will run. Next, the commerce rules will run (Hiding, then constraints, then validations). Finally, any advanced validations will run. Inactive rules will show up in italics in the left panel of the rule editor.

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""



Don’t use hiding rules in lieu of steps. Hiding rules should be used to make the page more dynamic to enhance usability. To hide attributes based on profiles, use steps instead of hiding rules.

RELATED TOPICS

COMMERCE DEPLOYMENT CENTER

OVERVIEW

Navigation: Admin Home Page > Commerce Processes > Process Definition > Deployment Center

Processes whose changes have not been deployed show a bold modification date on the Product Family List page. After a product family is deployed, the modification date displays in a normal font weight rather than bold; the last deployed date reflects the most recent deploy time. You can verify a successful deploy when you Review the confirmation email. A confirmation email is sent to the email address specified for a deploy; it indicates whether a deploy happened successfully or not. It also contains the details of what was deployed.

Event Types available for Scheduled Deployments

Event Type	Description
Deploy	This event ensures that a commerce process is ready for use on the commerce side.
Clone	This event creates a copy of the selected process on the admin side. The cloned process must be deployed for it to become active on the commerce-side. *Visible based on a commerce global setting under Admin Home Page > Global System Settings and Utilities > Commerce.*
Repopulate Column Data	This event makes it possible for changes done in the Data Columns on a deployed process, to be available on the commerce side. This event is applicable for the transaction manager data.
Diff	This event checks for differences between the last deploy and the commerce process and sends an email, if an email address is specified.
Remove Transactions	This event deletes all transactions from the commerce-side. *USE WITH CAUTION.*
Mass Update of Transactions	Set-up a BML script which can be used to update all existing commerce transactions. For example, if a sales rep quits their job and all of their existing quotes have to be reassigned to another sales rep, then the admin user could write a mass update BML script to replace the old sales rep's name with the new one. • Users can select the steps to which the Mass Update will be applied. • All transactions will be updated when the mass update script is run. • The process must be deployed before the mass update function is run. • This will be recorded in the document history as a System Mass Update action. You should find what you're looking for in the History XML. • The return type for this function is : document_number~variable_name~value[|document_number~variable_name~value]* • The admin side also provides a log that updates when the mass update function runs. Be careful when running mass update functions so that historical data in transactions is not improperly updated.

Deployment Status Messages

Status	Description
In Progress	The component is deploying. Deployment cannot be stopped once it has begun.
Pending	The component has been scheduled for deployment and is waiting in the queue. Until it moves to In Progress, the task can be deleted.
Loaded	The task has been requested but it has not yet been moved to the queue. You will move from Loaded to Pending and can delete the task if necessary.

ADMINISTRATION

Deploying a Commerce Process

1. Select the Process you want to deploy using the Select column.
2. Click **Open** under Quick Links. *A new window will open.*
3. Click **Deploy**.



You can also deploy a process be going to: Admin Home Page > Commerce Processes > Process Definition > Select a Process > Deployment Center.

Scheduling a Deployment

1. Under the **Send Confirmation Email To** section, select:
   - **Do not send email** and click **Add Event**.
   - **Email to** and enter the email address of the person you want to receive the deployment confirmation email and click **Add Event**.
   - Enter the desired date and time (in `mm/dd/yyyy hr:mm AM/PM` format) in the **Scheduled Time** field to schedule a deploy time. Click **Add Event**.
2. Click **Add Event** to display the name, status and the **Schedule Time** details for the new event. If desired, you can cancel the scheduled deployment by selecting the new event and clicking **Delete**. *You cannot remove an event if the deployment has already started.*
3. Click **Generate Schema and WSDL** to generate the Schema and WSDL.
4. Click **Refresh** to display the updated information on the page.

NOTES



When performing a Mass Update, users can filter which steps the update applies to.

Processes that have not been deployed are indicated with a blank timestamp in the Last Deployed column on the Process List page.

Deploys should take place when there is little activity on your site, like weekends or after-business



hours.



After a commerce process is deployed, some administrative functionality becomes hidden from view.

Deploy dates and times are recorded in the Last Deployed column on the Process List page.

RELATED TOPICS

COMMERCE ORDER OF OPERATIONS

OVERVIEW

New Transaction (variable name : _action_new_transaction)

- No Modify (tab) for this action
- No Advanced Modification for this action
- No Advanced Validation for this action
- No sub document exists at the point of quote creation

Order:

1. Default (tab) for all the main doc attributes
2. Main Doc Advanced Default

Open (variable name : _open_quote_process)

- No Advanced Modification for this action
- No Advanced Validation for this action

Order:

1. Modify tab

Copy (variable name : _action_process_resubmit)

- No Modify (tab) for this action
- No Advanced Modification for this action
- No Advanced Validation for this action

Order:

1. Create a new document based on the source document
2. Initialization (tab) for sub document
3. Initialization (tab) for main document
4. Sequence List

Email (variable name : _action_process_email_print)

- No Modify (tab) for this action
- No Advanced Modification for this action
- No Advanced Validation for this action
- Only Integration XSLs get executed

Print (variable name : _action_process_print)

- No Modify (tab) for this action
- No Advanced Modification for this action
- No Advanced Validation for this action
- Only Integration XSLs get executed

Forward (variable name : _action_process_forward)

- No Advanced Modification for this action
- No Advanced Validation for this action

Order:

1. Modify (tab) for sub document
2. Modify (tab) for main document

Add Line Items (User Define)

Order: From Commerce to Config

1. Modify (tab)
2. Advanced Modification
3. Advanced Validation

Order: From Config to Commerce

1. Sub Doc Advanced Default for each new line item
2. Attribute Default Function
3. Modify (tab) for main document

Browse (variable name : _browse_action)

- No Modify (tab) for this action
- No Advanced Modification for this action
- No Advanced Validation for this action

Order:

1. Invoke Auto fill Action when the user clicks “populate”

Reconfigure (variable name : _reconfigure_action)

Order: From Commerce to Config

1. Update Line Item – Modify (tab) for sub document
2. Reconfigure – Modify (tab) for main document
3. Reconfigure – Advanced Modification
4. Reconfigure – Advanced Validation

Order: From Config to Commerce

1. Sub Doc Advanced Default for each new line item
2. Update Line Item – Modify (tab) for sub document
3. Reconfigure – Modify (tab) for main document
4. Reconfigure – Advanced Modification
5. Reconfigure – Advanced Validation

Auto fill (variable name : _auto_fill_action)

Order:

1. Mappings (tab)
2. Modify (tab)
3. Advanced Modification
4. Advanced Validation

Remove (variable name : _remove_line_process)

Order:

1. Remove the selected line item(s)
2. Modify (tab)
3. Advanced Modification
4. Advanced Validation



order 1 & 4 – if the return value for the validation rule is not empty, or the simple validation rule for the "required" field (ex: data type, ranges, isnumber?, date format) are triggered, the line item(s) you selected for removal will not get removed. The line item(s) will show up again when you refresh the browser.

Select Alternate Address (variable name : _select_alternate_address_action)

Order: *(When the user clicks "Select This Entry")*

1. Mappings (tab)
2. Modify (tab)
3. Advanced Modification
4. Advanced Validation



It brings up a popup window with the alternate CRM address records

---

Update Line Items (variable name : _update_line_items)

Order:

1. Modify (tab) for sub document
2. Modify (tab) for main document
3. Advanced Modification
4. Advanced Validation

Copy Line Items (User Define)

Order:

1. Initialization (tab) (new line items only)
2. Update Line Item – Modify (tab) for sub document
3. Update Line Item – Modify (tab) for main document
4. Update Line Item – Advanced Validation
5. Recalculate (source line items)
6. Copy Line Items – Modify (tab)
7. Copy Line Items – Advanced Modification
8. Recalculate (copied line items)
9. Copy Line Items – Advanced Validation

Other User Defined Actions (Main Doc)

Order:

1. Update Line Items Advanced Modification
2. Main Doc Auto Update
3. Modify (tab)
4. Advanced Modification
5. Advanced Validation

Other User Defined Actions (Sub Doc)

Order:

1. Modify (tab)
2. Advanced Modification
3. Advanced Validation

NOTES



Main Doc Advanced Default – runs once when the document is created.

Sub Doc Advance Default – runs once on the new line item(s).

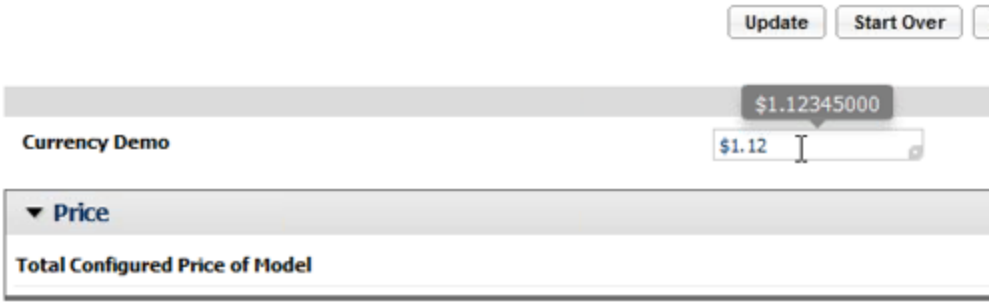
Integration scripts are executed in the order you specified under the Integration tab.

RELATED TOPICS

CURRENCY ATTRIBUTE HOVER TOOLTIP

OVERVIEW

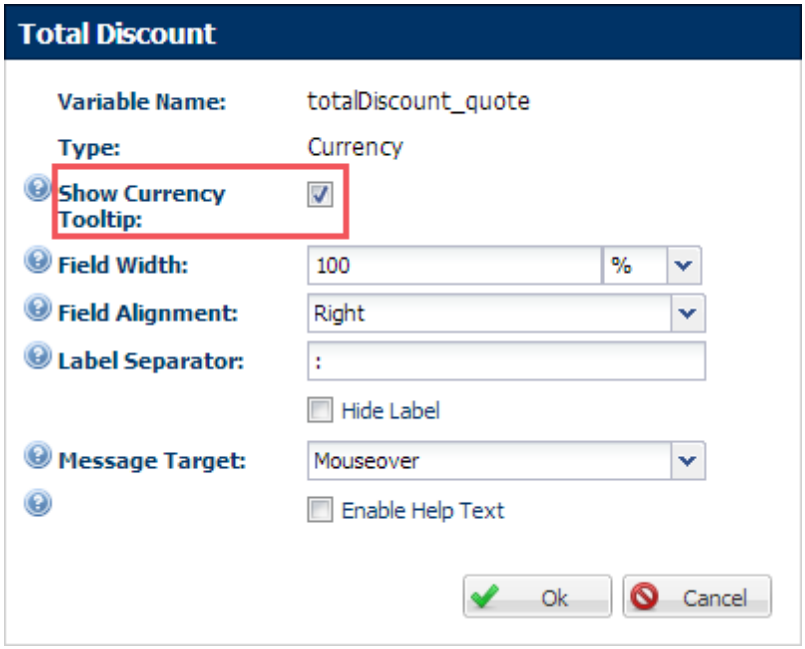
Enhanced decimal precision to includes a hover tooltip that will appear if the stored precision value is different than the displayed precision and the numbers are not equivalent. This functionality is available only on currency attributes



ADMINISTRATION

Enabling Currency Attribute Hover Tooltip in Commerce

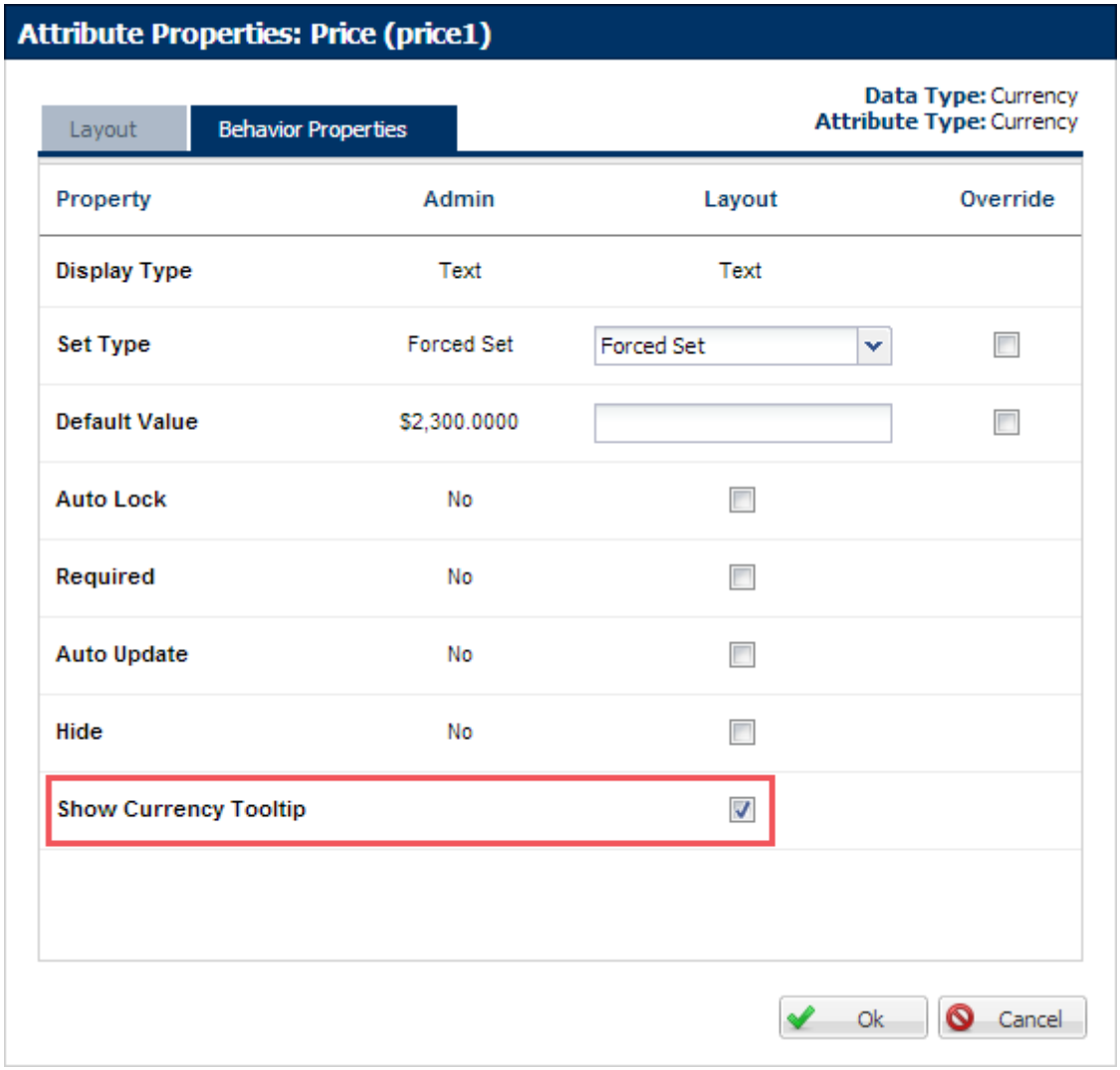
1. Navigate from: Admin Home Page > Process Definition Quick Links > Process > Select a Quote or Line> Select Layout Editor.
2. Click the edit icon on a currency attribute.
3. Select the **Show Currency Tooltip** option on the currency attribute editor.



For the Line Item Grid, the condition is for all attributes on the grid.

Enabling Currency Attribute Hover Tooltip in Configuration

1. Navigate from: Admin Home Page > Catalog Definition Quick Links > Product Family > Find List of Configuration Flow Rules > Select Layout.
2. Click the edit icon on a currency attribute.
3. Select the **Behavior Properties** tab.
4. Select the **Show Currency Tooltip** option.



This can also be done on the Behavior Properties tab at the grid level, if there are one or more currency attributes.

NOTES



Set default displayed precision, per currency, on the Host Company page.



Decimal precision is available for commerce and/or configuration attributes only, therefore the tooltip is only available on currency attributes.

Displayed precision cannot be greater than what is stored.

RELATED TOPICS

INVOKING A COMMERCE PROCESS

OVERVIEW

Process Invocation creates buttons that link product catalog pages with commerce processes. You can generate buttons that can either start a commerce process, open the Shopping Cart or move items to the Favorites List.

If commerce processes have already been created, you can generate process invocation buttons one at a time for each catalog page. Similar to other commerce components, process invocation buttons are created in an active, pending state.

Invocation Actions can be created on the following Catalog Pages:

Page List

Select	Page Title	Multiple/Single Types	Attributes	Last Deployed
☐	Part Search Results	Multiple	Parts	05/15/2013 12:23 PM
☐	Part Display	Single	Part	05/15/2013 12:23 PM
☐	Part Associations	Multiple	Parts	05/15/2013 12:23 PM
☐	Model Profile	Single	Model	05/15/2013 12:23 PM
☐	Model Configuration	Single	Model, Config	05/15/2013 12:23 PM
☐	Recommended Items	Single	Model, Config, Parts	05/15/2013 12:23 PM
☐	Favorites List	Multiple	Segments, Product Lines, Models, Parts	05/15/2013 12:23 PM
☐	Bill of Materials for Serial Number	Multiple	Parts	05/15/2013 12:23 PM
☐	Multiple Parts Buy	Multiple	Parts	05/15/2013 12:23 PM

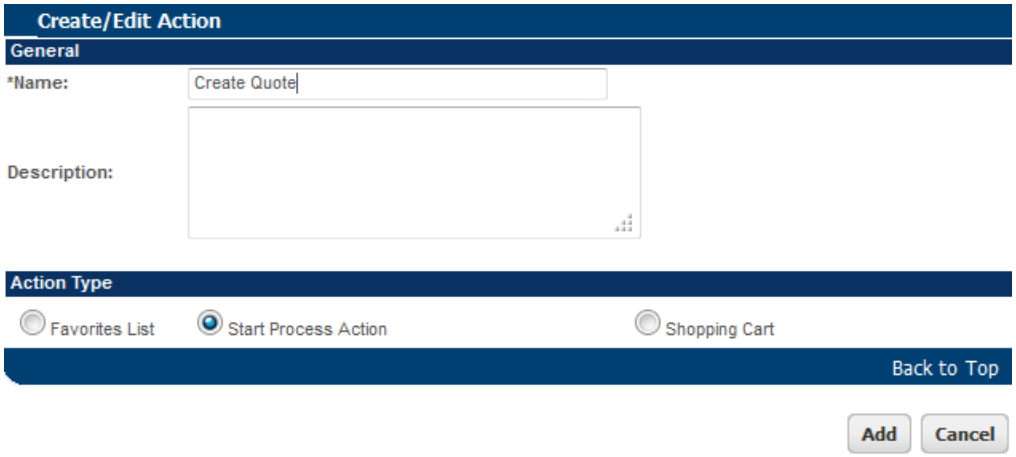
ADMINISTRATION

Adding an Invocation Action

1. Navigate from the Admin Home Page > Commerce and Documents > Process Invocation > Select a Catalog Page.
2. Click **Add**.
3. Enter a **Name** for the action that will display on the button on the user-side. *This is a required field and cannot be left blank.*
4. Enter a **Description** for the action (button).
5. Select an **Action Type**:
    - **Start Process Action** (Default): Clicking this button will invoke a commerce process, usually to create a new transaction.
    - **Shopping Cart:** Clicking this button will add items to or open the Shopping Cart.
    - **Favorites List:** Clicking this button will add products to a Favorites List.



The action type cannot be changed when editing the action.

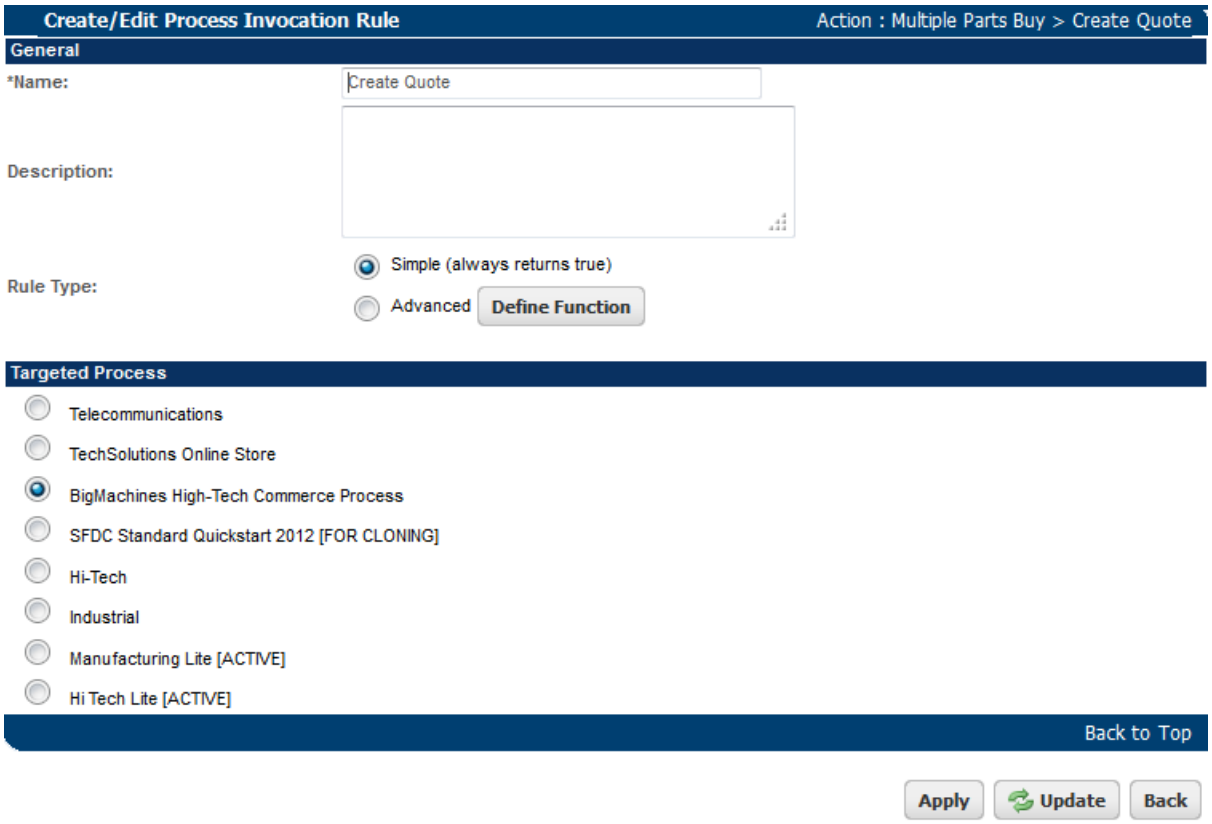


6. Click **Add**.

Adding or Editing an Invocation Rule

1. Navigate from the Admin Home Page > Commerce and Documents > Process Invocation > Click on a Catalog Page .
2. Choose **Rules** from the navigation drop-down menu of an action, then click **List**.
3. Click **Add.**
4. Enter a **Name** for the rule. *This is a required field and cannot be left blank.*
5. Enter a **Description** of the rule.
6. Select a **Rule Type**:
   - **Simple:** Simple rules will take users to a specific commerce process, which you'll choose in the next step.
   - **Advanced:** Click Define Functions to open a BML Editor and write a BML Script for the rule.
7. Select the radio button corresponding to the **target commerce process** you want to begin when the user clicks the action but-

ton.



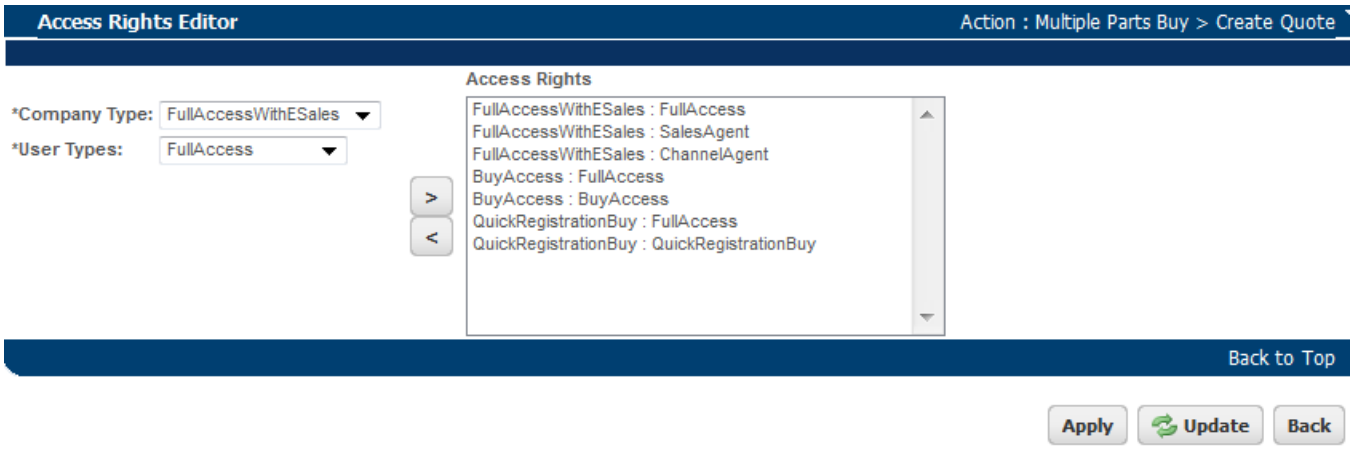
User Access Rights

1. Navigate from the Admin Home Page > Commerce and Documents > Process Invocation > Click on a Catalog Page .
2. Choose **User Access Rights** from the navigation drop-down menu of an action, then click **List**.
3. Choose a **Company Type and a** User Type from their respective drop-down menus.
4. Click the **(>**) arrow to add the new combination of Company Type and User Type in the Access Rights field. Repeat Steps 3 for multiple entries.



Click the (<) arrow to remove company type: user type combinations.

5. Click **Add**, Apply or Update to save changes.



Deploying Catalog Pages

1. Navigate from the Admin Home Page > Commerce and Documents > Process Invocation.
2. Use checkboxes to **select** all **catalog pages** that should be deployed.
3. Click **Deploy**.

NOTES

Buttons are created in an inactive, pending state. After a button is complete, meaning both user types and rules have been created, the button can be deployed to become active on catalog pages.

In order for a button to be visible outside of administration, the button must be deployed. The deploy process can be performed for individual catalog pages.

A commerce processes must already exist as they create the processes that the buttons invoke.

Action types cannot be changed after the button is deployed.

RELATED TOPICS

QUICK KEY LINE ITEMS

OVERVIEW

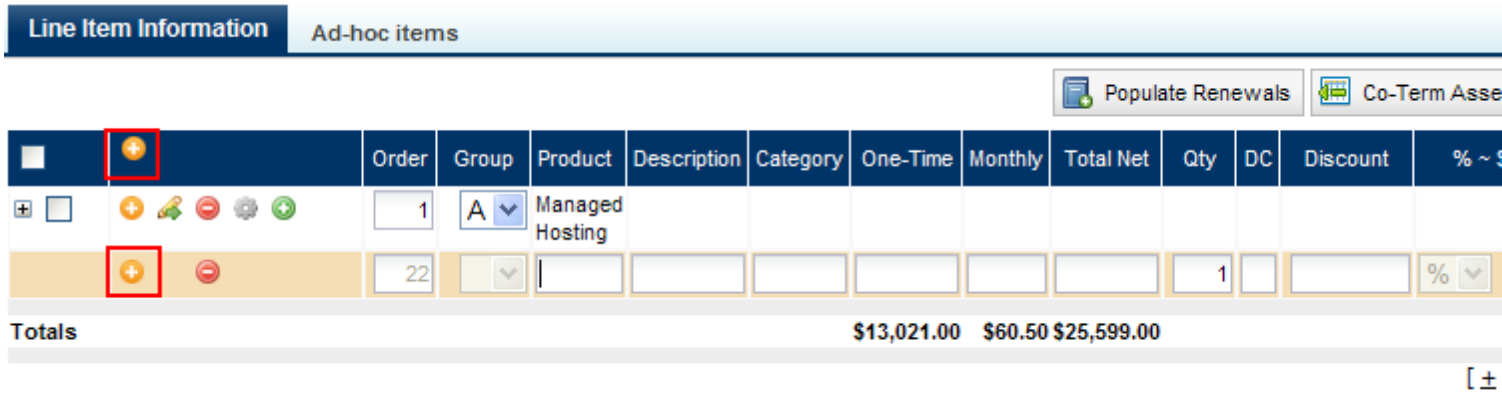
Quick key line items have the following capabilities:

- Add parts to a quote without navigating away from the transaction page, similar to an order-entry pad concept.
- Auto-suggest feature to aid in part selection.
- Default part fields in the line item grid populated using AJAX on tabbing out.

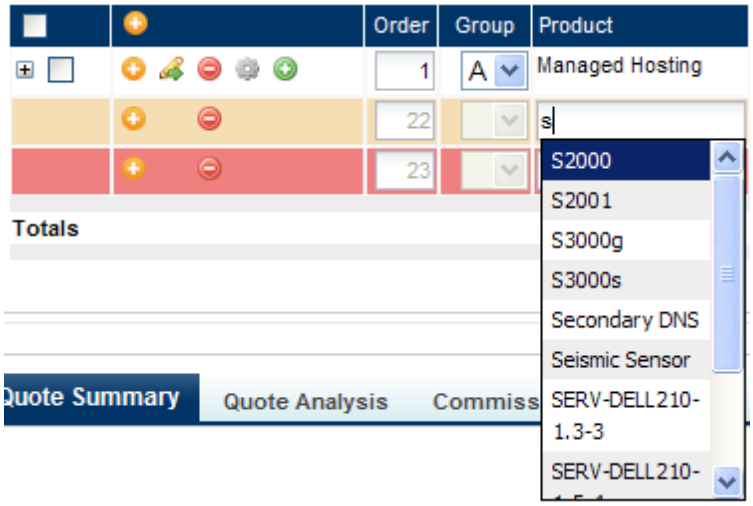
ADMINISTRATION

Adding Quick Key Line Items

1. Click the Quick Key Line Item icon.



To aid in part selection, an auto-suggest box appears as soon as you start typing in a part number. The parts suggested include Direct Buy parts that are available in the user's selected price book.



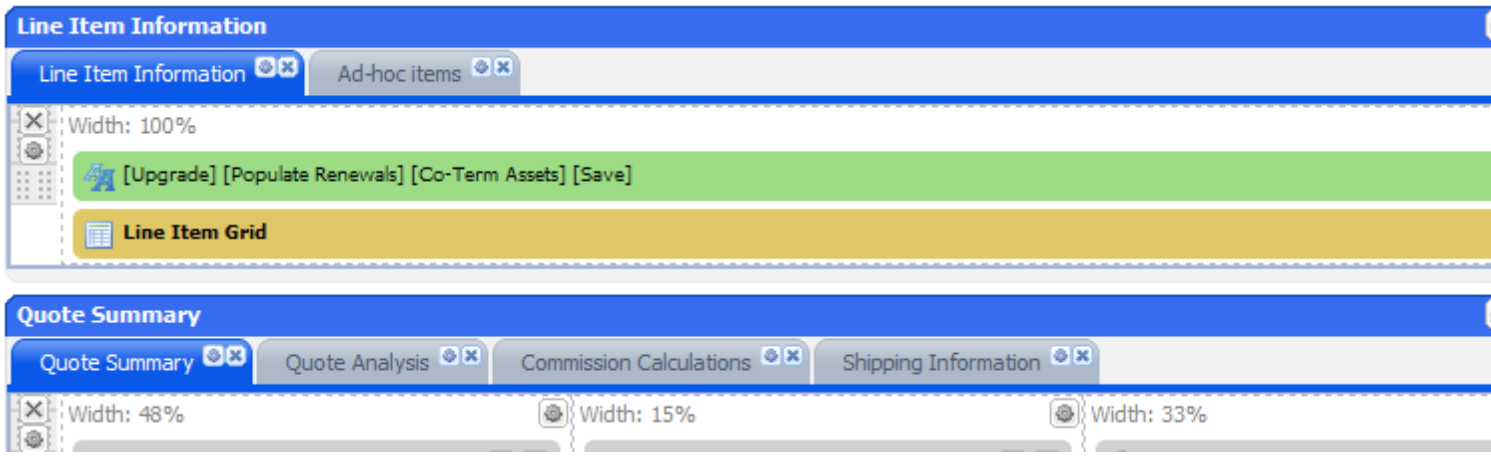
2. Edit the Part Number and Quantity. *If additional part fields are mapped to the line item grid, these fields will be automatically populated once the user enters a valid part number and tabs out of the part number field. Only part line items are supported.*

You must invoke a modify-type action to save all items added from the parts database.

Quick Key line items are added below the row in which the icon is clicked. They will be added below the entire configuration when the Quick Key icon is selected in the parent row.

Enabling Quick Key Line Items

1. Navigate to the Layout Editor of the commerce process for which you'd like to enable Quick Key Line Items.
2. Click the Gear icon on the Line Item Grid editor.



3. Select the **Display Quick Line Item** box within the Line Item Grid editor.
4. Ensure that the Part Number `(_part_number)` and Quantity `(_price_quantity)` fields are mapped to the line item grid.



You must ensure that a modify-type action exists for the commerce process since the end user is required to perform a modify-type action after adding Quick Key items. Use the commerce style-sheet to customize colors for Quick Key Items and any associated errors.

NOTES



If the user has access to multiple price books, the user must select a price book before adding items. If all Quick Key line items are deleted before savings, then the user will be asked to choose the price book again on a subsequent Quick Key action.



The Sequence Number field is non-editable unless the Quick Key line items are saved.

The access rights for Quick Key line items on a particular step is dependent upon those assigned to the main document Add from Catalog. The main document Add for Catalog must be either active or hidden for Quick Key line items to be available to the end user.

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""

RELATED TOPICS

USE CASE - RENEWALS

OVERVIEW

The renewals feature can take on many forms. A general use case can be implemented into the BigMachines commerce process to bring existing items from a previous sale to sell again for a new term. Using a repeatable scope, many different interpretations of the renewals feature can be accomplished by starting from a well worked out base template.

Prerequisites

Before using this tutorial, you must:

- Understand how to create commerce attributes and actions
- Be very familiar with BML
- Be able to navigate a commerce process
- Understand XSL's and the Pricing Script

ADMINISTRATION

Step 1: Creating Commerce Attributes

1. Navigate to the quote level of a commerce process.
2. Create the following main-document attributes:
    - **Renewals Holder String** (renewalsHolderString_quote); Text Area Attribute
    - **Co-Term Date** (coTermDate_quote); Date Attribute
3. Navigate to back to the Document List page and create the following sub-document attributes:
    - **Asset Start Date** (assetStartDate_line); Date Attribute
    - **Asset End Date** (assetEndDate_line); Date Attribute
    - **Asset ID** (assetID_line); Text Attribute
    - **Contract Start Date** (contractStart Date_line); Date Attribute
    - **Contract End Date** (contractEndDate_line); Date Attribute

---

Step 2: Creating Commerce Actions

1. Navigate to the Document List page.
2. Create the main-document modify action: **Add Renewals**.

3. Select the Advanced Validation option "**Modify Without Saving or Validating**", on the General Tab.
4. Create the main-document modify action Set Co-Term.

Step 3: Creating a Commerce Library Function

1. Navigate to the Document List page and create a commerce library called **Renewal Add Line Items**.
2. Click here to see BML Script.

```
/*
Quote > Add Renewal Line Items
Function: Takes in renewalsHolderString_quote and parses it out and makes webservice 1) add line items and 2) set attributes of said line items
Format: Asset1!!Asset2!!
Individual format: value1#value2#value3!!value1#value2#value3!!
*/
//Function will add items passed in to the desired quote
//remove the last "!!" from the string and turn into an array to loop over
itemArray = split(substring(renewalsHolderString_quote,0, len(renewalsHolderString_quote)-2), "!!"); bmiSessionId = _system_user_session_id;
siteName = lower(_system_company_name);
commerceProcess = "quickstart_commerce_process";
quoteProcess = "quote_process";
buyerCompanyName = "HT-QSDev";
NumOfAssets = 0; //will count how many assets we add dynamically
UpdatePartsXML = "";
//prep for addToTransaction API
addTransactionURL = "https://" + lower(siteName) +".bigmachines.com/bmfsweb/"+ lower(siteName) + "/image/WebServices/addToTransaction.xml";
soapString = urldatabyget(addTransactionURL,"","error");
soapString = replace(soapString, "SESSION_ID", bmiSessionId);
soapString = replace(soapString, "SITE_NAME", lower(siteName));
soapString = replace(soapString, "PROCESS_VAR_NAME", commerceProcess);
soapString = replace(soapString, "TRANSACTION_ID", transactionID_quote);
soapString = replace(soapString, "DOCUMENT_VAR_NAME_MAIN", quoteProcess);

itemsToAdd = "";
//Loop through items that are passed into function to add to quote
for item in itemArray {
//PartNumber = [0];
//Quantity = [1];
```

```
thisItem = split(item, "#");
itemsToAdd = itemsToAdd + "<bm:partItem>"
+ "<bm:part>" + thisItem[0]
+ "</bm:part>"
+ "<bm:quantity>" + substring(thisItem[1],0,find(thisItem[1],"."))
+ "</bm:quantity>"
+ "<bm:price_book_var_name>";
//+ thisItem[2]
//+ "_default_price_book" if(_quote_process_price_book_var_name <> "") {
itemsToAdd = itemsToAdd + _quote_process_price_book_var_name; }
else{ itemsToAdd = itemsToAdd +"_default_price_book"; }
itemsToAdd = itemsToAdd + "</bm:price_book_var_name>"
+ "</bm:partItem>";
NumOfAssets = NumOfAssets + 1;
}

soapString = replace(soapString , "ITEMS_TO_ADD", itemsToAdd);
//print soapString;
//soap call to add items
result = "";
result = urldatabypost("https://"+siteName +".bigmachines.com/v1_0/receiver", soapString, "");


//print result;


/* The result of adding line items will tell us the last document number. We can auto assume what the next document num-
bers will be by finding this document number that is returned
*/


ldnIndex1 = find(result,"<bm:last_document_number>");
ldnIndex2 = find(result,">",ldnIndex1);
ldnIndex3 = find(result,"</bm:last_document_number>");
ldnStr = substring(result,ldnIndex2+1,ldnIndex3);


//print ldnStr;
if (isnumber(ldnStr)) {
ldn = atoi(ldnStr);
//print ldn;
curDocNum = ldn - numOfAssets;
//print curDocNum;
}
```

```
else {
curDocNum = 2;
}
curSeqNum = 0;
updateTransactionURL = "https://" + lower(buyerCompanyName) +".bigmachines.com/bmfsweb/"+ lower(buyerCompanyName) +
"/image/WebServices/update_transaction.xml";
//print updateTransactionURL;
SoapUpdate = urldatabyget(updateTransactionURL,"","error");
//print "SOAPSTRING: " + soapString;
SoapUpdate = replace(SoapUpdate, "SESSION_ID", bmiSessionId);
SoapUpdate = replace(SoapUpdate, "SITE_NAME", lower(siteName));
SoapUpdate = replace(SoapUpdate, "PROCESS_VAR_NAME", commerceProcess);
SoapUpdate = replace(SoapUpdate, "TRANSACTION_ID", transactionID_quote);
SoapUpdate = replace(SoapUpdate, "BUYER_COMPANY_NAME", buyerCompanyName);
SoapUpdate = replace(SoapUpdate, "SUPPLIER_COMPANY_NAME", _system_supplier_company_name);
SoapUpdate = replace(SoapUpdate, "DOCUMENT_VAR_NAME_MAIN", quoteProcess);
SoapUpdate = replace(SoapUpdate, "ACTION_VAR_NAME", "_update_line_items");
SoapUpdate = replace(SoapUpdate, "LOGIN_NAME", _system_user_login);
//Get sequence number that is required for updateTransaction API
for line in line_process{
if(line._sequence_number > curSeqNum) {
curSeqNum = line._sequence_number;
}
}
for item in itemArray {
//Asset Start Date(previous) = [2]
//Asset End Date(previous)/Auto Set Asset Start Date (new) = [3]
//Asset SFDC ID = [5]
curDocNum = curDocNum + 1;
curSeqNum = curSeqNum + 1;
thisItem = split(item, "#");
UpdatePartsXML = UpdatePartsXML + "<bm:line_process bm:bs_id='" + transactionID_quote + "' bm:buyer_company_name='" +
buyerCompanyName + "' bm:buyer_user_name='superuser' bm:currency_pref='USD' bm:data_type='3' bm:document_name='Line'
bm:document_number='" + string(curDocNum) + "' bm:document_var_name='line_process' bm:process_var_name='quickstart_com-
merce_process' bm:supplier_company_name='" + buyerCompanyName + "'>"
+ "<bm:_document_number>" + string(curDocNum) + "</bm:_document_number>"
+ "<bm:_sequence_number>" + string(curSeqNum) + "</bm:_sequence_number>"
+ "<bm:assetStartDate_line>" + thisItem[2]+ " 00:00:00" + "</bm:assetStartDate_line>"
+ "<bm:assetEndDate_line>" + thisItem[3]+ " 00:00:00" + "</bm:assetEndDate_line>"
+ "<bm:assetId_line>" + thisItem[5] + "</bm:assetId_line>";
```

```
//Do calculation for base contractEndDate on contractTermMenu_quote
if(thisItem[3] <> "") {
contractEndDate = strtojavadate(thisItem[3],"yyyy-MM-dd");
contractStartDate = adddays(strtojavadate(thisItem[3],"yyyy-MM-dd"),1);
if(contractTermMenu_quote == "12") {
contractEndDate = adddays(contractStartDate,365);
}
UpdatePartsXML = UpdatePartsXML + "<bm:contractEndDate_line>" + datetostr(contractEndDate,"yyyy-MM-dd HH:mm:ss") +
"</bm:contractEndDate_line>"
+ "<bm:contractStartDate_line>" + datetostr(contractStartDate,"yyyy-MM-dd HH:mm:ss")+ "</bm:contractStartDate_line>";

}
UpdatePartsXML = UpdatePartsXML + "</bm:line_process>";

}

SoapUpdate = replace(SoapUpdate, "LINE_PROCESS", UpdatePartsXML);
//print SoapUpdate;
update_result = urldatabypost("https://"+siteName +".bigmachines.com/v1_0/receiver", SoapUpdate, "");
//print update_result;
return "";
```

Step 4: File Manager

1. Navigate to the File Manager and create a Web Services folder. Here you will store two XML files:

- update_transaction.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>SESSION_ID</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://SITE_NAME.bigmachines.com/bmfsweb/SITE_NAME/schema/v1_0/commerce/PROCESS_VAR_NAME.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:updateTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:transaction>
<bm:id>TRANSACTION_ID</bm:id>
<bm:process_var_name>PROCESS_VAR_NAME</bm:process_var_name>
<bm:buyer_company_name>BUYER_COMPANY_NAME</bm:buyer_company_name>
<bm:supplier_company_name>SUPPLIER_COMPANY_NAME</bm:supplier_company_name>
<bm:data_xml>
<bm:DOCUMENT_VAR_NAME_MAIN bm:bs_id="TRANSACTIONID_quote" bm:buyer_company_name="BUYER_COMPANY_NAME" bm:buyer_user_name="LOGIN_NAME" bm:data_type="0" bm:document_name="Quote" bm:document_number="1" bm:-document_var_name="DOCUMENT_VAR_NAME_MAIN" bm:process_var_name="PROCESS_VAR_NAME" bm:supplier_company_name-e="SUPPLIER_COMPANY_NAME">
<bm:sub_documents>
LINE_PROCESS
</bm:sub_documents>
</bm:DOCUMENT_VAR_NAME_MAIN>
</bm:data_xml>
<bm:action_data>
<bm:action_var_name>ACTION_VAR_NAME</bm:action_var_name>
<bm:performer_name>LOGIN_NAME</bm:performer_name>
<bm:performer_type>fullAccess</bm:performer_type>
<bm:performer_company_name>BUYER_COMPANY_NAME</bm:performer_company_name>
</bm:action_data>
</bm:transaction>
```

```
</bm:updateTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

- addToTransaction.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>SESSION_ID</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://SITE_NAME.bigmachines.com/bmfsweb/SITE_NAME/schema/v1_0/commerce/PROCESS_VAR_NAME.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:addToTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:items>
ITEMS_TO_ADD
</bm:items>
<bm:transaction>
<bm:process_var_name>PROCESS_VAR_NAME</bm:process_var_name>
<bm:id>TRANSACTION_ID</bm:id>
<bm:action_data>
<bm:document_var_name>DOCUMENT_VAR_NAME_MAIN</bm:document_var_name>
</bm:action_data>
</bm:transaction>
<bm:include_transaction>true</bm:include_transaction>
</bm:addToTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

Step 5: Integration XSL's

1. Create two XSL files on your computer:
    - Integration XSL Generator

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:-
fo="http://www.w3.org/1999/XSL/Format" version='1.0'>
<xsl:output method="xml"/>
<xsl:variable name="main_doc" select="/transaction/data_xml/document[@data_type=0]"/>
<xsl:strip-space elements="*"/>
<xsl:template match="/">
<!-- Begin SOAP XML -->
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:x-
si="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Header>
<SessionHeader xmlns="urn:partner.soap.sforce.com">
<sessionId><xsl:value-of select="/transaction/user_info/session_id"/></sessionId>
</SessionHeader>
<CallOptions xmlns="urn:partner.soap.sforce.com">
<client>BigMachinesLFE/1.0</client>
</CallOptions>
</soap:Header>
<soap:Body>
<query xmlns="urn:partner.soap.sforce.com">
<queryString>
select Asset.Id, Asset.Name,Asset.Quantity, Asset.InstallDate, Asset.UsageEndDate, Asset.Status, Pro-
duct2.Id, Product2.BigMachines__Part_Number__c FROM Asset WHERE AccountId ='<xsl:value-of select-
t="normalize-space($main_doc/_customer_id)"/>'
</queryString>
</query>
</soap:Body>
</soap:Envelope>
<!-- End SOAP XML -->
</xsl:template>
</xsl:stylesheet>
```

    - XSL Integration Parser

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:soapen-
v="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sf2="urn:partner.soap.sforce.com" xmlns:s-
f="urn:sobject.partner.soap.sforce.com">
<xsl:variable name="main_doc" select="/transaction/data_xml/document[@data_type=0]"/>
<xsl:output method="xml"/>
<xsl:template match="*">
<data_xml>
<document document_var_name="quote_process">
<renewalsHolderString_quote>
<xsl:for-each select="//sf2:records">
<xsl:value-of select=".//sf:BigMachines__Part_Number__c" />
<xsl:text>#</xsl:text>
<xsl:value-of select="./sf:Quantity" />
<xsl:text>#</xsl:text>
<xsl:value-of select="./sf:InstallDate"/>
<xsl:text>#</xsl:text>
<xsl:value-of select="./sf:UsageEndDate"/>
<xsl:text>#</xsl:text>
<xsl:value-of select="./sf:status"/>
<xsl:text>#</xsl:text>
<xsl:value-of select="./sf:Id"/>
<xsl:text>!!</xsl:text>
</xsl:for-each>
</renewalsHolderString_quote>
</document>
</data_xml>
</xsl:template>
</xsl:stylesheet>
```

2. Navigate to the **Integration XSL Files page** for the commerce process.
3. Create a **new Integration XSL**.
4. Select **Integration Type Import** (Query Generator and Result Parser).
5. Name the Integration XSL: **SFDC Assets - Import** (sfdcAssetsImport).
6. Enter **quote_process.none** into the ID Field.
7. Browse for the Integration XSL Generator file you created and select it for the SOAP Generator XSL file.
8. Browse for the Integration XSL Parser file you created and select it for the Results Parser XSL file.

9. Click **Add**.

Putting It All Together

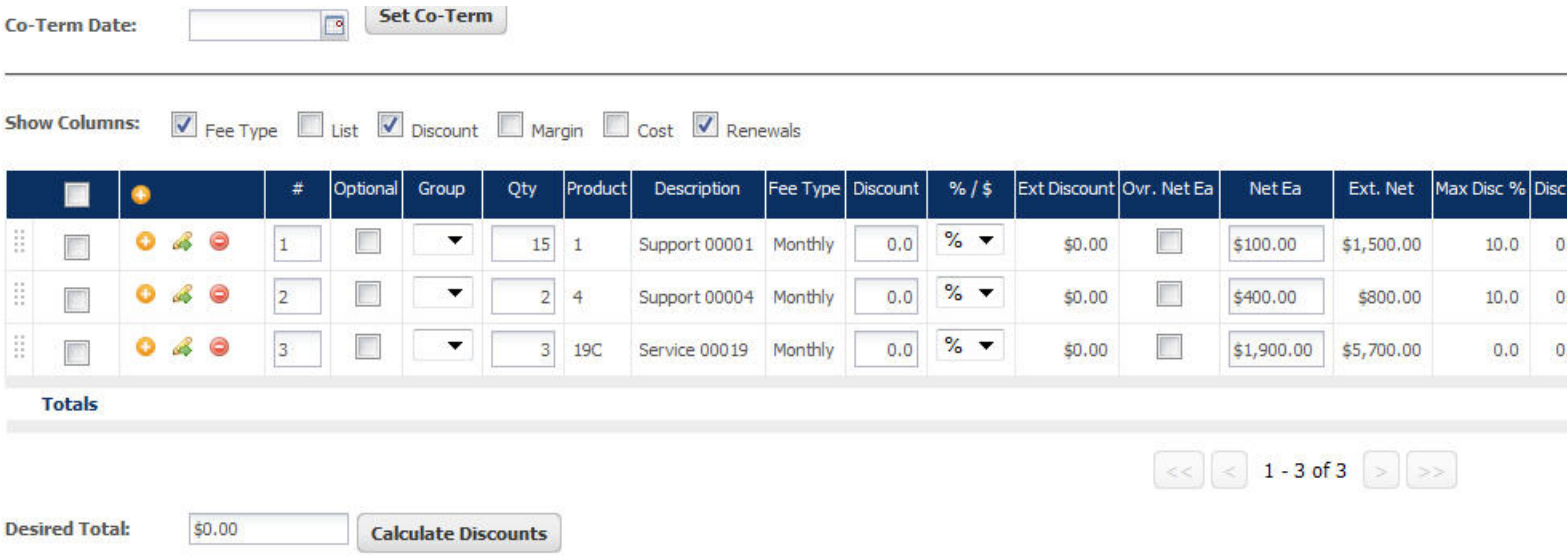
The integration XSL file that pulls in assets will run whenever a quote is opened. It stores the asset information in a delimited string within the renewalsHolderString_quote attribute. The format of the delimited string is:

```
bmiPartNumber#Qty#InstallDate#UsageEndDate#status#AssetID!!
```

The Add Renewals button can be clicked only during the Pending steps. This action then invokes the commerce library function Renewal Add Line Items. This will take the renewalsHolderString_quote and parse it out to add line items to the quote. It will also update the attributes through 2 webservice calls addToTransaction and updateTransaction. The destination redirects to the same page to display the newly added line items and 'refresh' the page.

Line items are now added to the quote based on clicking the Add Renewals button. You can start terming items using the contract start and end date attributes.

Co term is set at global level. The co term date attribute is dispersed to all monthly items and sets that date to ALL recurring line items. The co term action runs the pricing script with the 'coterm' as the action. The pricing function has been adapted to set the contract end dates of all recurring items and use that date in recalculating the price.



NOTES

In the requirements, we require the prices of recurring items to be set as monthly prices. To figure out the daily price, the calculation is as follows:
LineRecurringPrice = (price*12)/365 ---> Essentially
Pricing totals will be in one-time fees, monthly recurring and total contract value. The monthly totals will include all recurring items as 'what you would expect to see as 1 month pricing of your items.' With co-



terming and daily prices, the total contract prices will capture exact prices down to the day.



We recommend your administrators are an Advanced Blue Belt or beyond to implement this solution. This should only be adopted if you have a strong technical team.

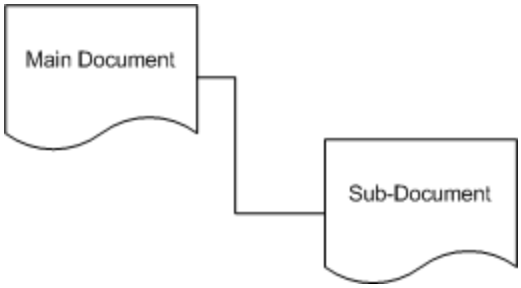
RELATED TOPICS

DOCUMENTS OVERVIEW

OVERVIEW

Each commerce process contains one or more document sets. There are two types of commerce documents: **main document *(quote)*** and **sub document *(line)***. A document set is comprised of a main document or a main document with an attached sub-document. Using these document types, you can create flexible commerce processes. **Example:** a process named RFQ could contain three main documents (RFQ, Quote and PO), plus three corresponding sub-documents (RFQ sub, Quote sub, PO sub). Main documents are used for general information pertaining to the transaction, like customer name, billing and shipping information, and sub-documents are used as product detail pages.

For every process, commerce documents are created in a one-to-one relationship: *every main document can be mapped to only one sub document.*



Commerce documents permit a one-to-many relationship, where the main document is the `r;one`, and the sub-document is the `r;many`. This design scheme enables buyers to create a single main document for multiple products, each of which appears on a linked sub-document where users can view product details and configure products separately. A summary of sub-document data also appears on main documents, in the requisite `line item` group. You can create transitions between documents using actions. Transitions can be created between main and sub-documents using the system-generated document action, `r;open sub-document`. Transitions can be created between two main documents using workflow steps.

Attributes and **actions** are the building blocks of documents. Attributes are used to create document fields, whereas actions trigger events to happen. Along with the custom attributes and actions you can create, the commerce system provides a set of **standard attribute sets** and **system-generated actions**. After generating document attributes and actions, all the pieces can be arranged using **document views**. Only those attributes and actions included in a view appear to users.

Group Sequence Number

End users can view recommended items and child line items with sequence numbers that are counted based on the parent line item's sequence.This functionality is enabled when you add the `_group_sequence_number` attribute to the attribute list and the line item grid instead of `_sequence_number`. End users can order line items through drag and drop when the sequence number/group sequence number is present on the line item grid. **Example:** a parent line item can have a group sequence number of 1 and then the child items will have group sequence number of 1.1, 1.2, 1.3...etc.



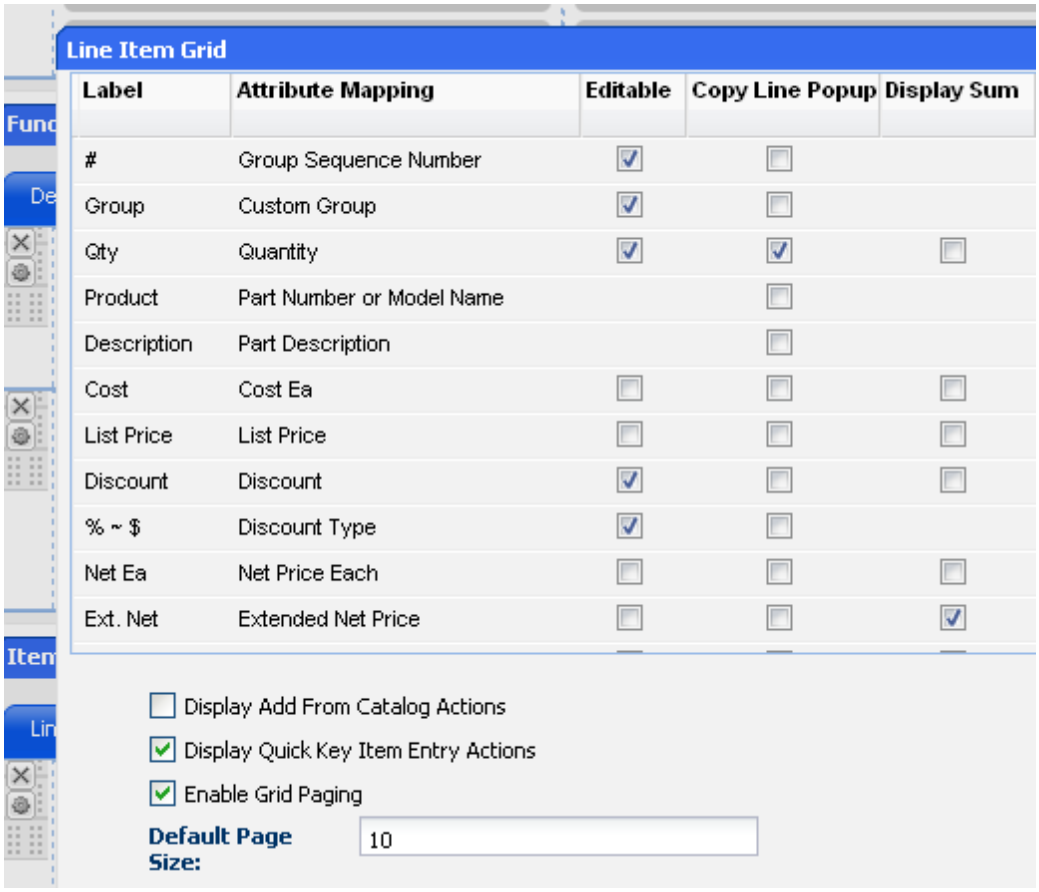
#	Group	Qty	Product	Description	Cost	List Price	Ext List	Discount	% ~ $	Ext Discount	Ovr. Net Ea	Net Ea
1			Mid Sized SUV									
1.1		1	F25X0040	F25X0040	$0.00	$743.75	$743.75	0.0	%	$0.00		$743.75
1.2		1	F25X	F25X	$0.00	$35,516.25	$35,516.25	0.0	%	$0.00		$35,516.25

Enabling Group Sequence Numbers and Drag & Drop Line Items

1. Verify there is a Group Sequence Number sub-document (line) attribute with the variable name `_group_sequence_number`.



2. Add the **Group Sequence Number** to the Line item Grid on the Main Document Layout Editor.



Group Sequence Number functionality is independent of Sequence Number. *It should not be used in conjunction with Sequence Number.*
Line items are automatically re-sequenced if the user enters a value in the group sequence number field. However, if the user enters an invalid value, then the user's input is ignored.

User-Defined Document Actions

Action Name	Description
Modify	This is the most common type of action and is used to change attribute values, modify quote data and save modifications to the transaction.
Back	This is an action that takes the user back tot he last viewed page. It does not save user data.
Add from Catalog	This sends the user to the home page to select items to add to the transaction.

Action Name	Description
Print	Displays a document created by the Document Engine. It is modify type action and hence possesses the same characteristics.
Email	Emails a document created by the Document Engine.
Display History	This is used to display the change history for a document. This action requires an XSL format file that controls how and which changes appear.
Export Attachment	This enables a user to add the proposal (PDF) to the related partner Opportunity (only on an integrated site). It possesses the same characteristics as a modify-type action.
Copy to Favorites *	This is a special action that is used to copy selected line items from the transaction to the favorites. From here the user can add it back to the transaction.
Copy Line Items *	This enables users to make one or more copies of one or more line items (configured and non-configured) at a time. It's a modify type action; however, unlike other modify type actions, the Copy Line Item Action possesses an Initialization tab and has no Integration tab.
Submit *	This creates a special set of Approval Actions and Attributes used to set-up approval sequences.

***** Represents a system-generated action that is available on the main-document only.

System-Generated Document Actions

Action Name	Description
Update Line Items	Enables users to modify line item data on a main document and then save changes.
Remove [sub-document]	Enables users to remove products.
Open [sub-document]	Enables users to view product details on a sub-document.
Create [next main-document]	Enables users to move on to the next main document in the commerce process. *The **Create** button automatically appears with all main documents.*
Reconfigure *	Enables users to change a product's configuration. *The system automatically creates a **Reconfigure** button when a Configurable Attribute Set (which is a composite attribute) is added to a main document.*
Recalculate *	Enables user to recalculate pricing data after a pricing field is altered. *The **Recalculate** action is generated automatically when a Price Attribute Set is added to a sub-document.*
Create [sub-document] *	Enables users to create another sub-document. *The system automatically creates the **Create** [sub-document] action if there is another sub-document attached to the next main document in order.*
Autofill/Browse/View/Select Alternate Address	These are generated automatically when a Customer ID attribute field is added to a main document. They are also automatically removed from a main-document if the Customer ID attribute is removed. **Autofill:** a user can enter an ID into the Customer ID attribute, click the Autofill action and automatically link data from the Accounts database tot he main document. Its variable name is `_auto_fill_action`. To trigger a default autofill with user or account information, set this option on the process administration page. **Browse:** enables user to search the Accounts database for a customer record. Its variable name is `_browse_action`. **View:** enables suppliers to view an Accounts profile when an ID appears in the Customer ID attribute. It's variable name is `_view_crm_action`. **Select Alternate Address:** enables host company to associate an alternate Accounts address with a commerce document. The system only displays this action when the Customer ID associated with a commerce document has multiple addresses. Its variable name is `_select_alternate_address_action`.

***** Represents a system-generated action that is available on the main-document only.

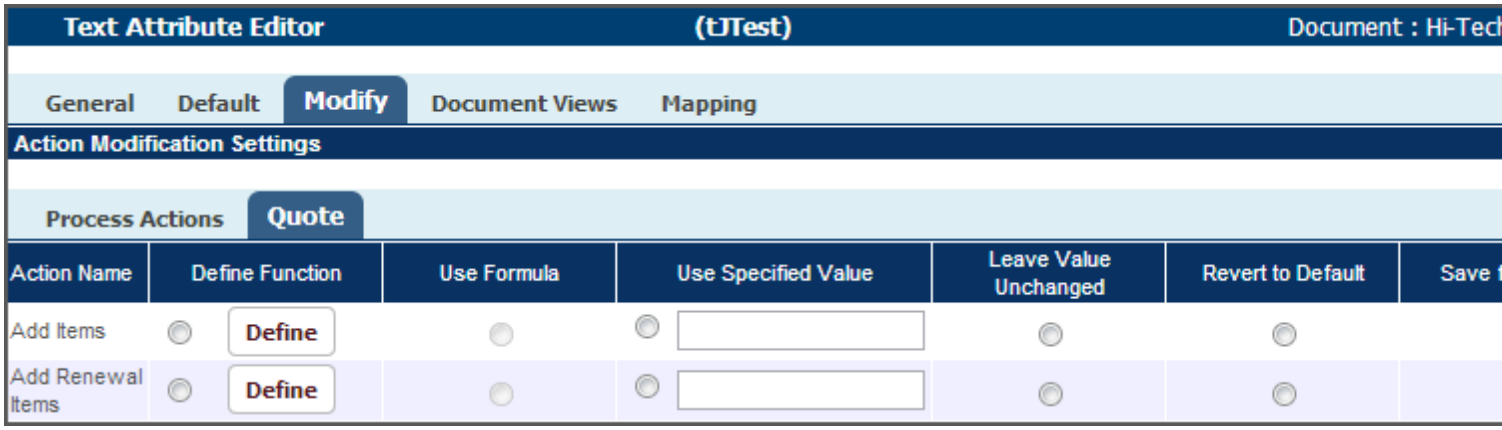
The system automatically deletes system-generated actions when their corresponding attribute or attribute set is deleted if attribute sets are associated.

ADMINISTRATION

Managing Documents

The Documents page helps you in managing commerce documents within the selected process.

1. Click **Add** to add a main or a sub-document.
2. Select a **Document Name**. *This displays the Add Document page for adding and editing the document.*
3. Select an option from the **Navigation** drop-down list and click **List**. This displays the corresponding screens for the option selected. *The options provided are:*
   - **Attributes:** Clicking **List** against **Attributes** displays the **Attributes List** page for managing attributes for documents within the selected process.
   - **Actions:** Clicking **List** against **Actions** displays the **Actions List** page.
   - **Groups:** Clicking **List** against **Groups** displays the **Group List** page.
4. Administrators can select which quote level formulas will be executed when a new line item is added to a quote on the Modify tab within the Attribute Editor.



Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line

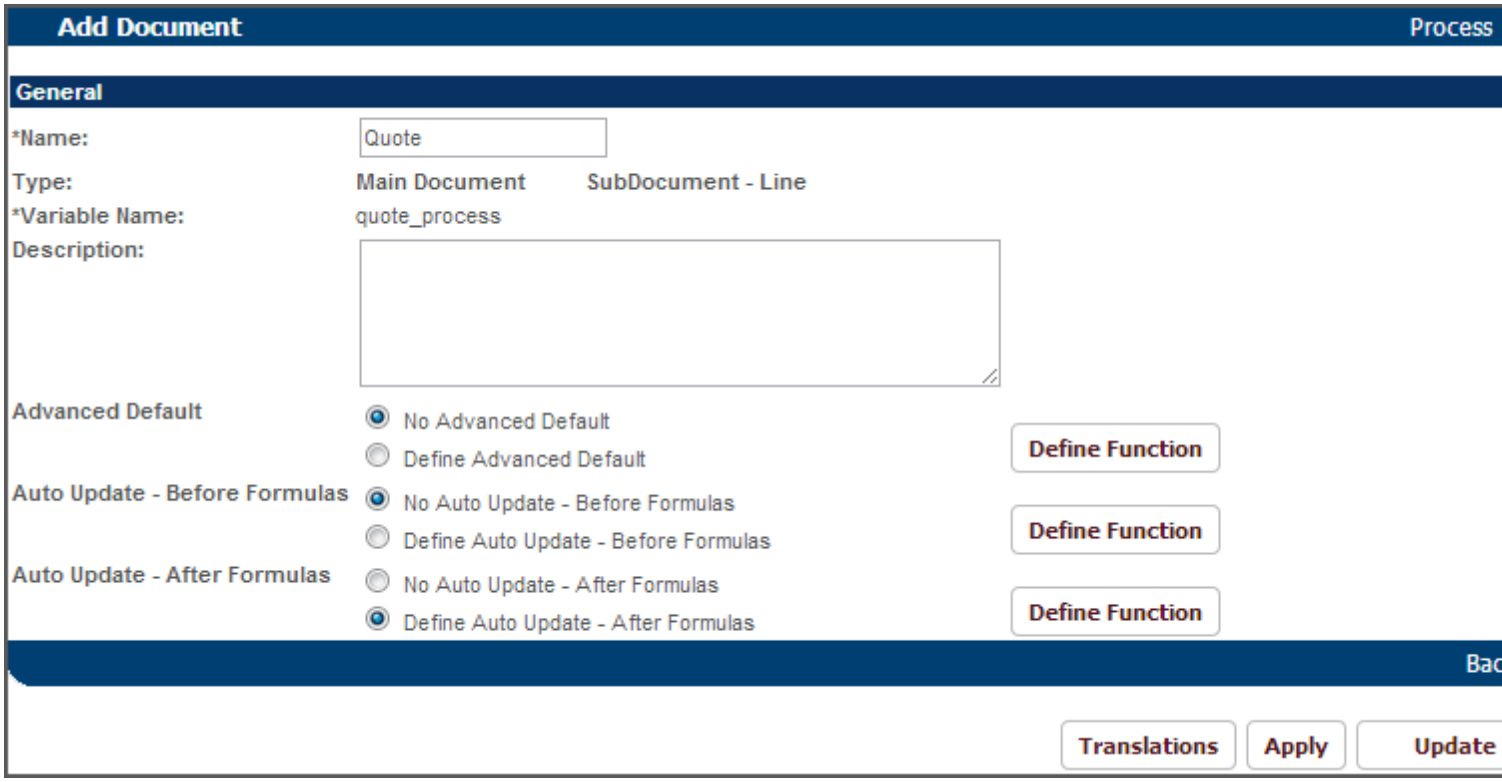
Items.

5. Click **Back** to return to the Processes page.

Adding a Main Document (quote level)

Documents can be created one at a time within a commerce process.

1. Enter a **Name**. This label distinguishes one document from another and is used as the page title on the commerce side whenever a user opens the document type. *This is a required field.*
2. Choose the document type as **Main** or **Sub-document**. *It is recommended that you create your sub-documents first.*
3. Enter a unique **Variable Name** for your reference. *This is a required field.*



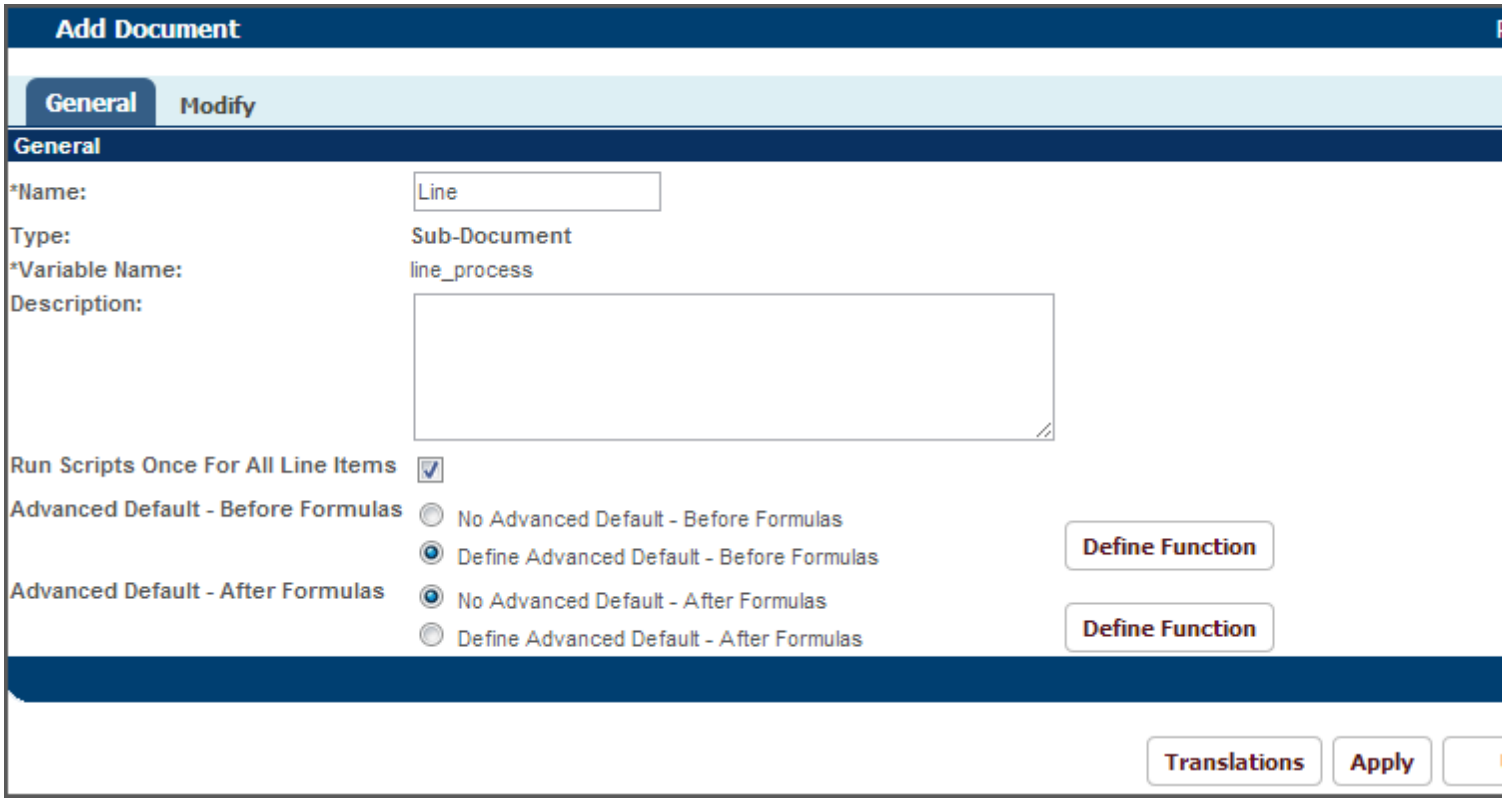
4. Enter a document **Description**.
5. **Advanced Default:** Either select the **No Advanced Default** if you do not want to have an advanced default, or select **Define Advanced Default** and click **Define Function**. *This displays the **Select Attributes** window where you can select the **System Variable Name**, **Variable Name for (Quote)**, **Variable Name for (Line)**, and/or **Library Function(s)** for the selected document. Clicking **Next Screen** takes you to the **Function Editor** page.*

6. **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.
7. **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*
8. Click one of the following actions:
    - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
    - **Apply:** saves changes and remains on the page.
    - **Update:** saves changes and returns to the Document List page.
    - **Back:** returns to the Document List page without saving changes.

Adding a Sub-Document (line level)

Documents can be created one at a time within a commerce process.

1. Enter a **Name**. This label distinguishes one document from another and is used as the page title on the commerce side whenever a user opens the document type. *This is a required field.*
2. Enter a unique **Variable Name** for your reference. *This is a required field.*
3. Enter a document **Description**.

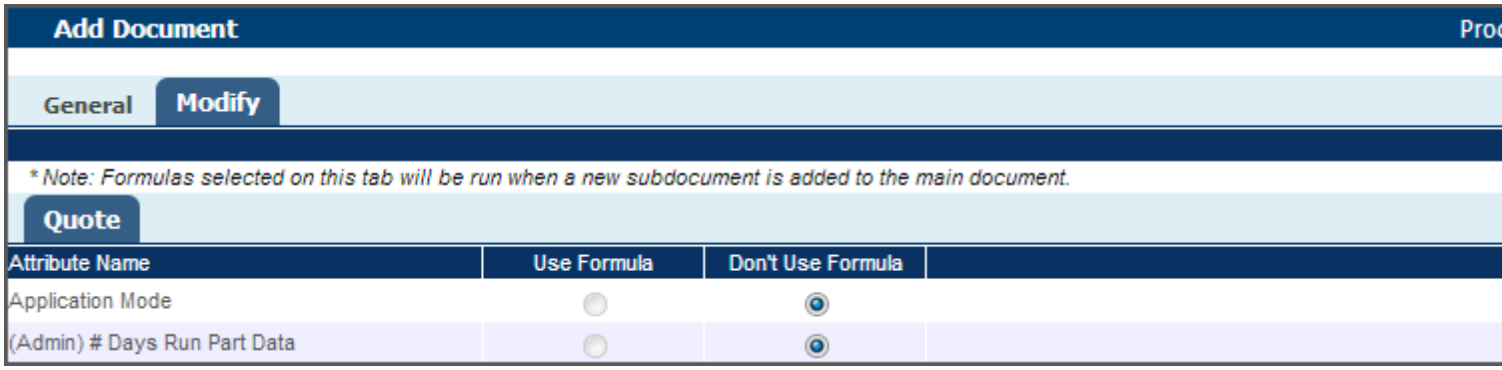


4. Select whether to **Run Scripts Once For All Line Items**.

5. **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.

6. **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*

7. Click the **Modify** tab to choose whether to **Use Formula** or **Don't Use Formula** for certain quote-level attributes. *Only quote level attributes exist on the line level Modify tab. For each of the quote level attribute the user can specify whether to use or not use formulas during a new line item addition.*



8. Click one of the following actions:
    - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
    - **Apply:** saves changes and remains on the page.
    - **Update:** saves changes and returns to the Document List page.
    - **Back:** returns to the Document List page without saving changes.

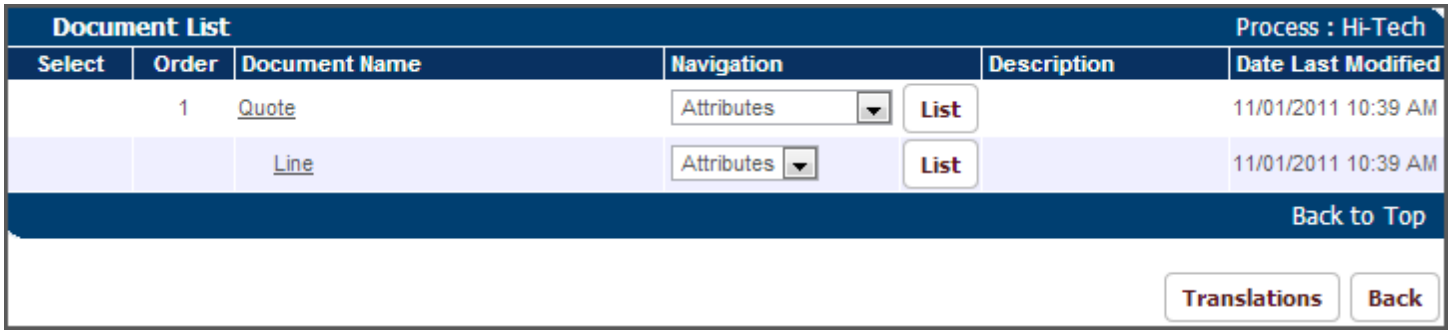


Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items.

Order of execution: Modify tab, Advanced Default- Before Formulas, Formulas, and Advanced Default- After Formulas.

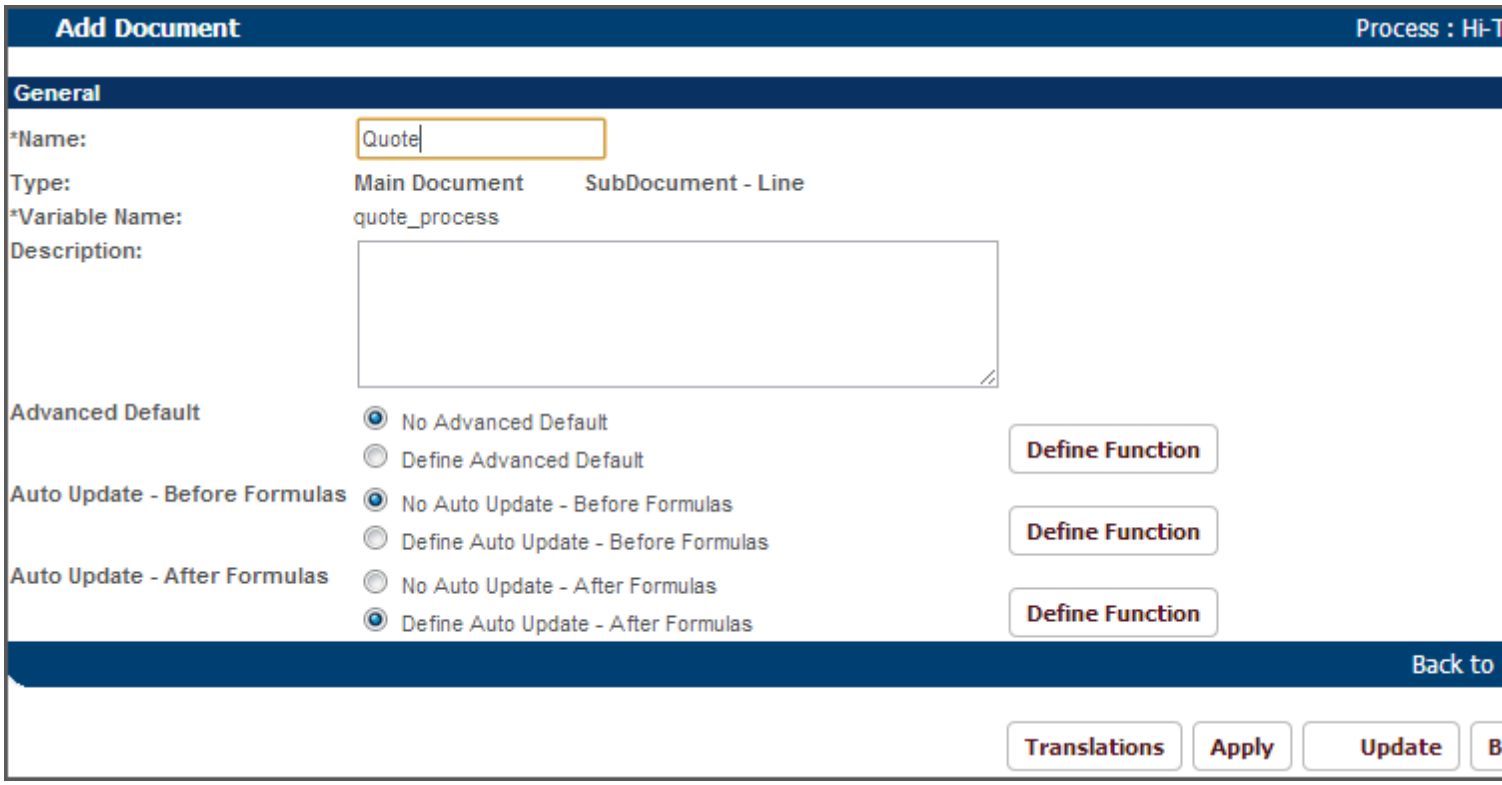
Editing a Main Document (quote level)

1. Click the name of the document you want to edit on the **Document List** page.



2. Edit the **Name** and/or **Description** displayed on the Add Document page.
3. Edit the following information, if necessary:
    - **Advanced Default:** Either select the **No Advanced Default** if you do not want to have an advanced default, or select **Define Advanced Default** and click **Define Function**. *This displays the **Select Attributes** window where you can select the **System Variable Name**, **Variable Name for (Quote)**, **Variable Name for (Line)**, and/or **Library Function (s)** for the selected document. Clicking **Next Screen** takes you to the **Function Editor** page.*
    - **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.

- **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*



4. Select one of the following options:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Apply:** saves changes and remains on the page.
   - **Update:** saves changes and returns to the Document List page.
   - **Back:** returns to the Document List page without saving changes.



Document names and descriptions can be edited at any time.

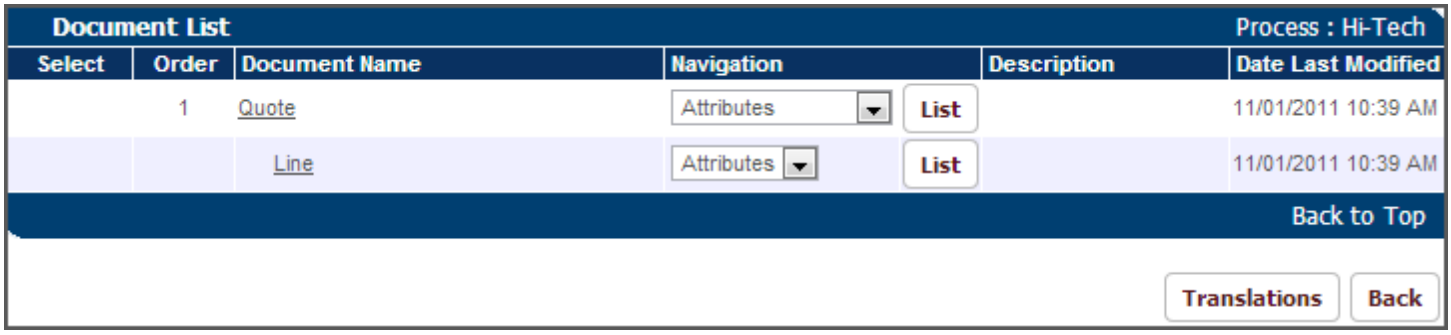


Quote level (main-document) formulas can be executed when users add a line item (sub-document) to

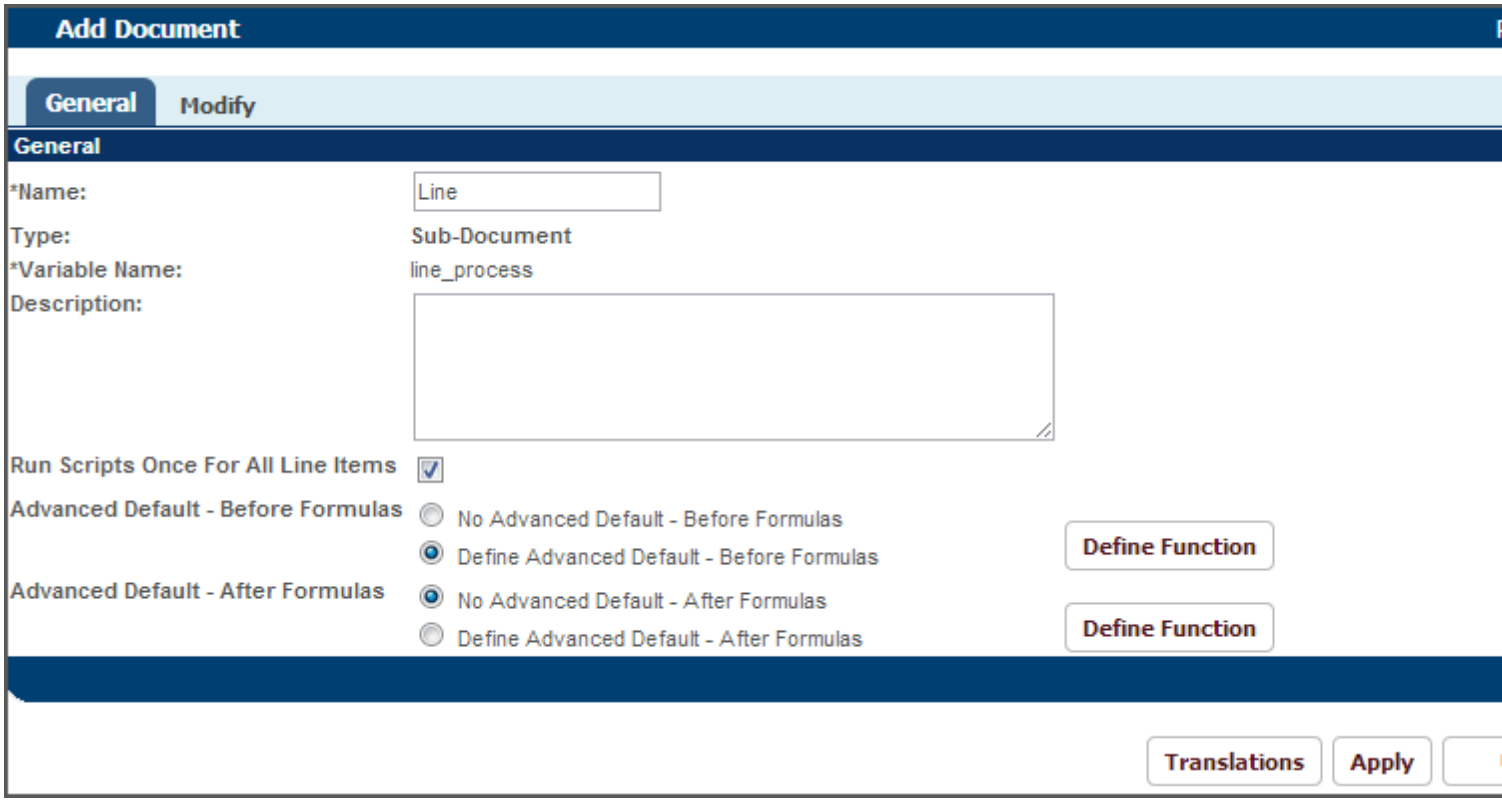
the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items.

Editing a Sub-Document (line level)

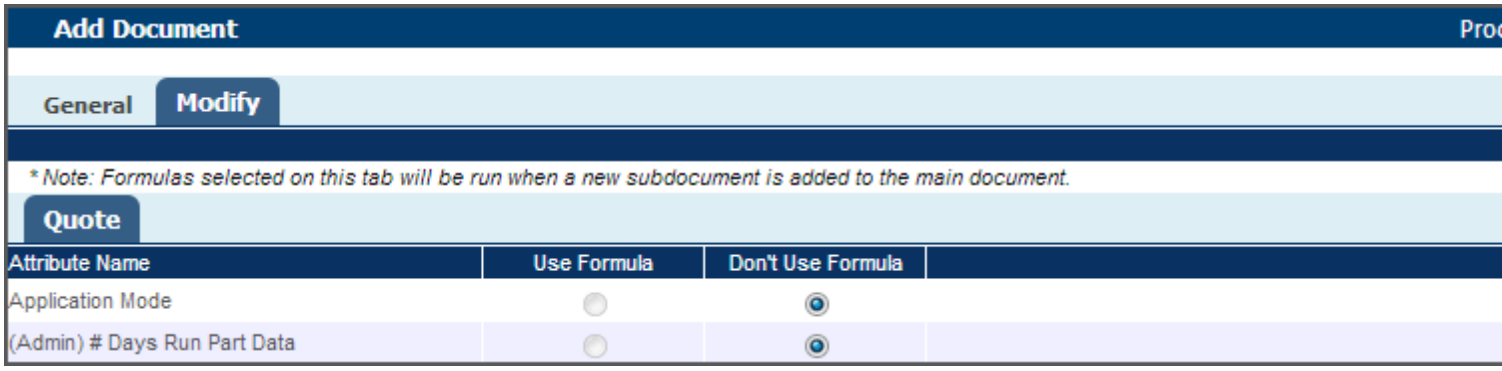
1. Click the name of the document you want to edit on the **Document List** page.



2. Edit the **Name** and/or **Description** displayed on the Add Document page.
3. Edit the following information, if necessary:
   - Select whether to **Run Scripts Once For All Line Items**.
   - **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.
   - **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*
   - Click the **Modify** tab to choose whether to **Use Formula** or **Don't Use Formula** for certain quote-level attributes.



4. Click the **Modify** tab to choose whether to **Use Formula** or **Don't Use Formula** for certain quote-level attributes. *Only quote level attributes exist on the line level Modify tab. For each of the quote level attribute the user can specify whether to use or not use formulas during a new line item addition.*



5. Select one of the following options:

   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Apply:** saves changes and remains on the page.
   - **Update:** saves changes and returns to the Document List page.
   - **Back:** returns to the Document List page without saving changes.



Document names and descriptions can be edited at any time.



Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items.

Order of execution: Modify tab, Advanced Default- Before Formulas, Formulas, and Advanced Default- After Formulas.

Deleting a Document

Use caution when deleting a commerce document because it is not possible to recover the deleted data. Upon a deletion, all associated views, attributes and actions within the document are also deleted.

1. Select the checkbox for the corresponding document you wish to delete on the Document List page.
2. Click **Delete**.

Ordering a Document

Ordering documents to determine how they appear in the **Commerce Manager**. Order numbers are auto-generated at the time of creating documents. As only one sub-document is associated with a main document, sub-documents don't require ordering.

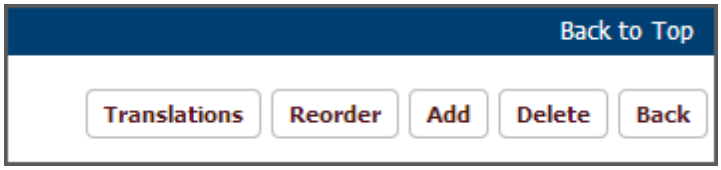
1. On the **Document List** page, click on the order number against the document and set it to any positive integer. *The order numbers of other main documents in the list will be automatically reordered.*
2. Click **Update**.



Changes to document order deletes current document mapping data.

Ordering Main & Sub-Documents

1. Navigate from: Document List > Quote or Line Attributes > List
2. Scroll to bottom of page and click **Reorder**.



3. Enter a new order number for the attributes you wish to reorder.

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*
        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*
    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



Edits are not visible until the process is deployed (or redeployed).

Create main documents as the primary documents in a set. Create sub-documents as detail pages that link up with main documents, that is, Line Item page for a RFQ.

There is an option "Run Once For All Line Items" next to the advanced default on sub-documents. Standard functionality of the Advanced Default is that it will run every time a line is added to a quote. Having this option enabled enhances performance.



Previously created variable names and previously set document types appear read-only because they are not editable.



Sub-documents and Main documents have a one-to-one relationship, meaning only one sub-document can be associated with a main document.

Main document Advanced Default will run when the transaction is first created and its purpose is to set main doc attribute default values. The purpose is to have one script that sets all of these attribute values as once, as opposed to pulling each of them separately. This also allows you to ease maintenance as all the defaults are set in a single script.

Sub-document (line item) Advanced Default will run whenever a line items is added to a quote. The purpose of writing this script is to set all of the line-item attribute values at once.



You cannot add, delete or update order numbers for documents in a deployed commerce process.

Certain areas of document administration are only visible for processes that have not been deployed. You cannot delete documents once a process is deployed.

User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

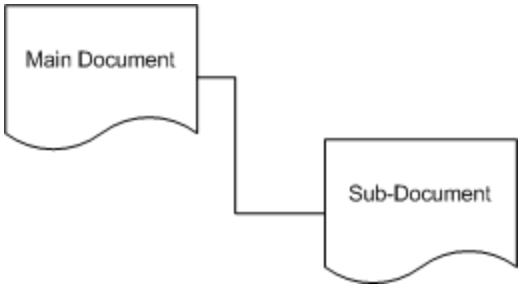
RELATED TOPICS

DOCUMENTS OVERVIEW

OVERVIEW

Each commerce process contains one or more document sets. There are two types of commerce documents: **main document *(quote)*** and **sub document *(line)***. A document set is comprised of a main document or a main document with an attached sub-document. Using these document types, you can create flexible commerce processes. **Example:** a process named RFQ could contain three main documents (RFQ, Quote and PO), plus three corresponding sub-documents (RFQ sub, Quote sub, PO sub). Main documents are used for general information pertaining to the transaction, like customer name, billing and shipping information, and sub-documents are used as product detail pages.

For every process, commerce documents are created in a one-to-one relationship: *every main document can be mapped to only one sub document.*



Commerce documents permit a one-to-many relationship, where the main document is the `r;one`, and the sub-document is the `r;many`. This design scheme enables buyers to create a single main document for multiple products, each of which appears on a linked sub-document where users can view product details and configure products separately. A summary of sub-document data also appears on main documents, in the requisite `line item` group. You can create transitions between documents using actions. Transitions can be created between main and sub-documents using the system-generated document action, `r;open sub-document`. Transitions can be created between two main documents using workflow steps.

Attributes and **actions** are the building blocks of documents. Attributes are used to create document fields, whereas actions trigger events to happen. Along with the custom attributes and actions you can create, the commerce system provides a set of **standard attribute sets** and **system-generated actions**. After generating document attributes and actions, all the pieces can be arranged using **document views**. Only those attributes and actions included in a view appear to users.

Group Sequence Number

End users can view recommended items and child line items with sequence numbers that are counted based on the parent line item's sequence.This functionality is enabled when you add the `_group_sequence_number` attribute to the attribute list and the line item grid instead of `_sequence_number`. End users can order line items through drag and drop when the sequence number/group sequence number is present on the line item grid. **Example:** a parent line item can have a group sequence number of 1 and then the child items will have group sequence number of 1.1, 1.2, 1.3...etc.



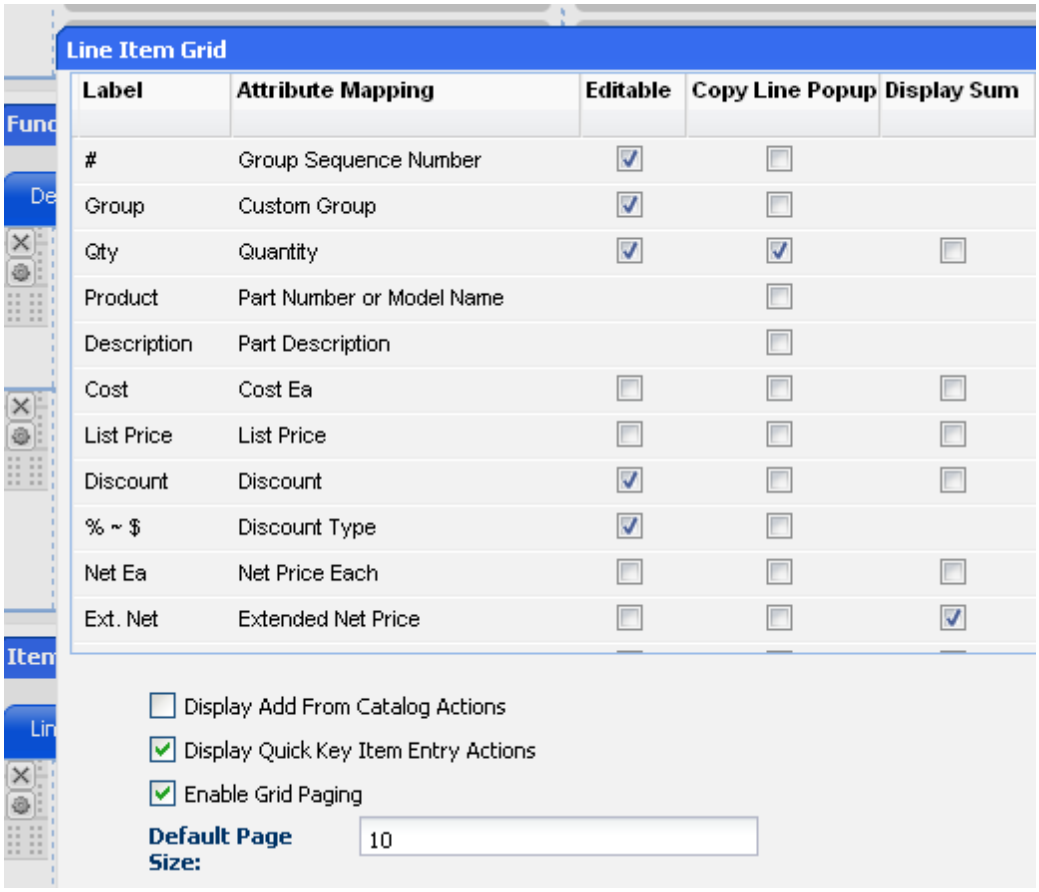
#	Group	Qty	Product	Description	Cost	List Price	Ext List	Discount	% ~ $	Ext Discount	Ovr. Net Ea	Net Ea
1			Mid Sized SUV									
1.1		1	F25X0040	F25X0040	$0.00	$743.75	$743.75	0.0	%	$0.00		$743.75
1.2		1	F25X	F25X	$0.00	$35,516.25	$35,516.25	0.0	%	$0.00		$35,516.25

Enabling Group Sequence Numbers and Drag & Drop Line Items

1. Verify there is a Group Sequence Number sub-document (line) attribute with the variable name `_group_sequence_number`.



2. Add the **Group Sequence Number** to the Line item Grid on the Main Document Layout Editor.



Group Sequence Number functionality is independent of Sequence Number. *It should not be used in conjunction with Sequence Number.*
Line items are automatically re-sequenced if the user enters a value in the group sequence number field. However, if the user enters an invalid value, then the user's input is ignored.

User-Defined Document Actions

Action Name	Description
Modify	This is the most common type of action and is used to change attribute values, modify quote data and save modifications to the transaction.
Back	This is an action that takes the user back tot he last viewed page. It does not save user data.
Add from Catalog	This sends the user to the home page to select items to add to the transaction.

Action Name	Description
Print	Displays a document created by the Document Engine. It is modify type action and hence possesses the same characteristics.
Email	Emails a document created by the Document Engine.
Display History	This is used to display the change history for a document. This action requires an XSL format file that controls how and which changes appear.
Export Attachment	This enables a user to add the proposal (PDF) to the related partner Opportunity (only on an integrated site). It possesses the same characteristics as a modify-type action.
Copy to Favorites *	This is a special action that is used to copy selected line items from the transaction to the favorites. From here the user can add it back to the transaction.
Copy Line Items *	This enables users to make one or more copies of one or more line items (configured and non-configured) at a time. It's a modify type action; however, unlike other modify type actions, the Copy Line Item Action possesses an Initialization tab and has no Integration tab.
Submit *	This creates a special set of Approval Actions and Attributes used to set-up approval sequences.

***** Represents a system-generated action that is available on the main-document only.

System-Generated Document Actions

Action Name	Description
Update Line Items	Enables users to modify line item data on a main document and then save changes.
Remove [sub-document]	Enables users to remove products.
Open [sub-document]	Enables users to view product details on a sub-document.
Create [next main-document]	Enables users to move on to the next main document in the commerce process. *The **Create** button automatically appears with all main documents.*
Reconfigure *	Enables users to change a product's configuration. *The system automatically creates a **Reconfigure** button when a Configurable Attribute Set (which is a composite attribute) is added to a main document.*
Recalculate *	Enables user to recalculate pricing data after a pricing field is altered. *The **Recalculate** action is generated automatically when a Price Attribute Set is added to a sub-document.*
Create [sub-document] *	Enables users to create another sub-document. *The system automatically creates the **Create** [sub-document] action if there is another sub-document attached to the next main document in order.*
Autofill/Browse/View/Select Alternate Address	These are generated automatically when a Customer ID attribute field is added to a main document. They are also automatically removed from a main-document if the Customer ID attribute is removed. **Autofill:** a user can enter an ID into the Customer ID attribute, click the Autofill action and automatically link data from the Accounts database tot he main document. Its variable name is `_auto_fill_action`. To trigger a default autofill with user or account information, set this option on the process administration page. **Browse:** enables user to search the Accounts database for a customer record. Its variable name is `_browse_action`. **View:** enables suppliers to view an Accounts profile when an ID appears in the Customer ID attribute. It's variable name is `_view_crm_action`. **Select Alternate Address:** enables host company to associate an alternate Accounts address with a commerce document. The system only displays this action when the Customer ID associated with a commerce document has multiple addresses. Its variable name is `_select_alternate_address_action`.

***** Represents a system-generated action that is available on the main-document only.

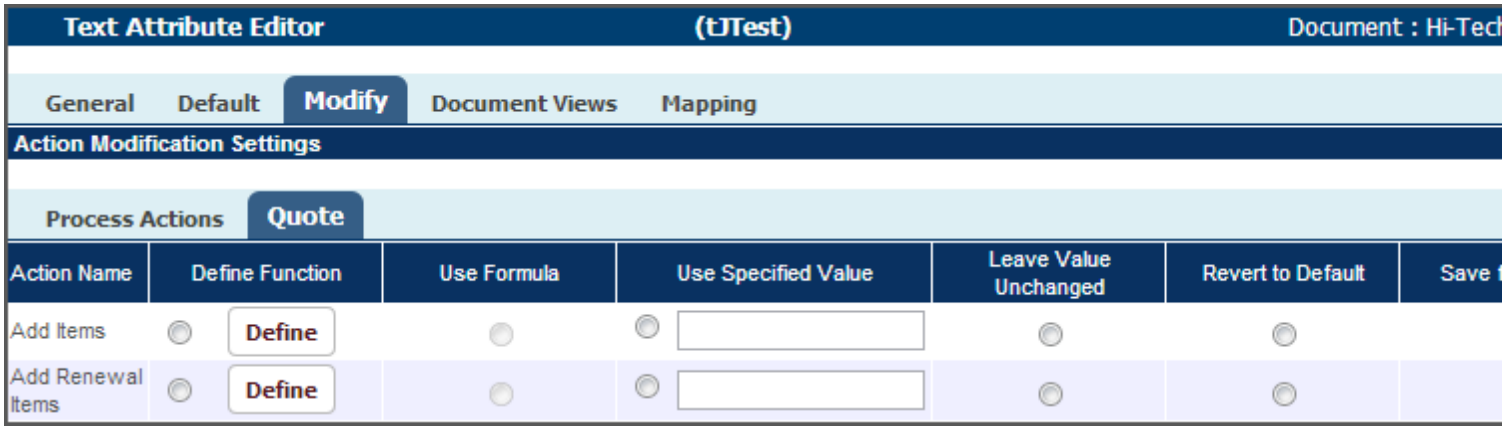
The system automatically deletes system-generated actions when their corresponding attribute or attribute set is deleted if attribute sets are associated.

ADMINISTRATION

Managing Documents

The Documents page helps you in managing commerce documents within the selected process.

1. Click **Add** to add a main or a sub-document.
2. Select a **Document Name**. *This displays the Add Document page for adding and editing the document.*
3. Select an option from the **Navigation** drop-down list and click **List**. This displays the corresponding screens for the option selected. *The options provided are:*
   - **Attributes:** Clicking **List** against **Attributes** displays the **Attributes List** page for managing attributes for documents within the selected process.
   - **Actions:** Clicking **List** against **Actions** displays the **Actions List** page.
   - **Groups:** Clicking **List** against **Groups** displays the **Group List** page.
4. Administrators can select which quote level formulas will be executed when a new line item is added to a quote on the Modify tab within the Attribute Editor.



Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line

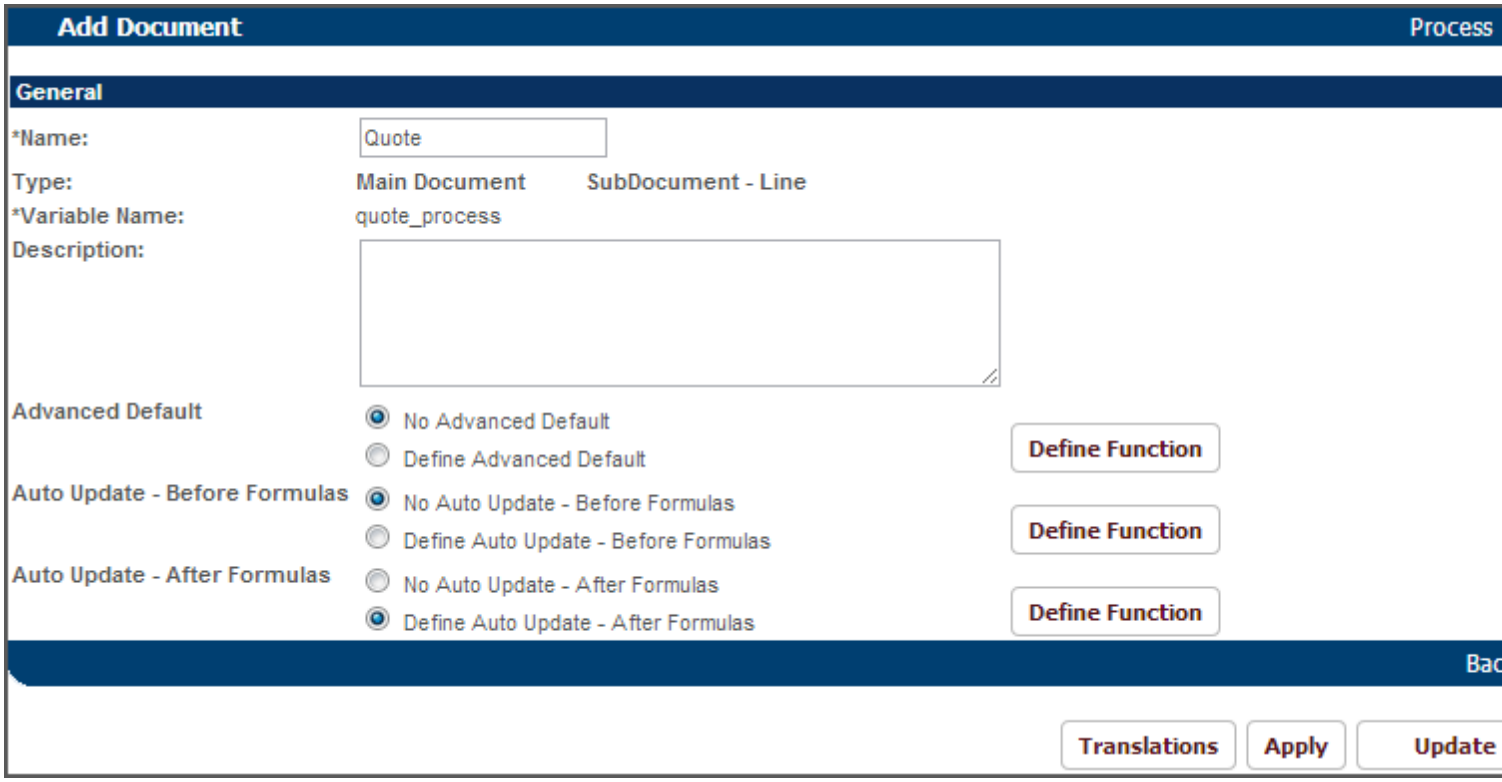
Items.

5. Click **Back** to return to the Processes page.

Adding a Main Document (quote level)

Documents can be created one at a time within a commerce process.

1. Enter a **Name**. This label distinguishes one document from another and is used as the page title on the commerce side whenever a user opens the document type. *This is a required field.*
2. Choose the document type as **Main** or **Sub-document**. *It is recommended that you create your sub-documents first.*
3. Enter a unique **Variable Name** for your reference. *This is a required field.*



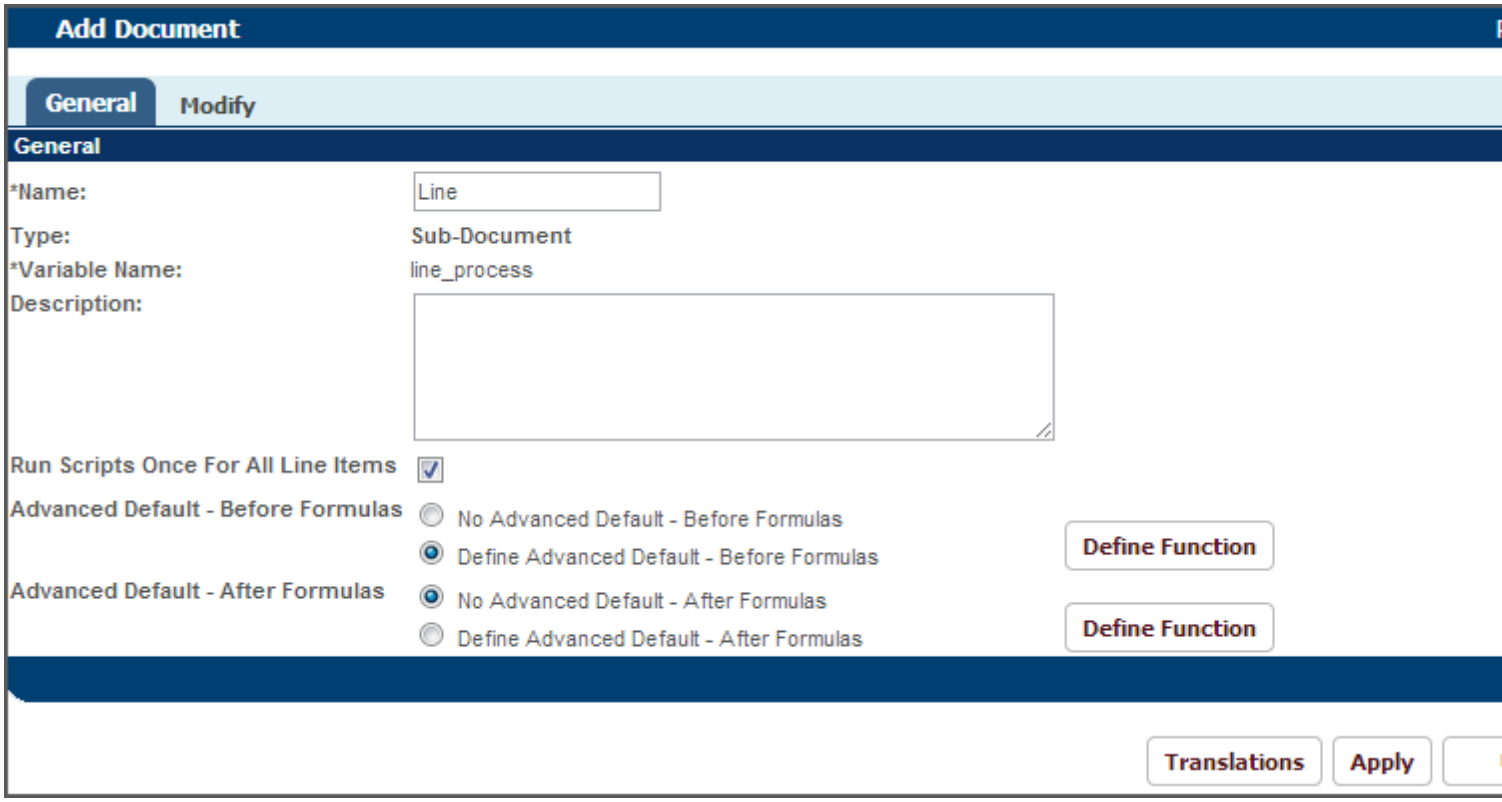
4. Enter a document **Description**.
5. **Advanced Default:** Either select the **No Advanced Default** if you do not want to have an advanced default, or select **Define Advanced Default** and click **Define Function**. *This displays the **Select Attributes** window where you can select the **System Variable Name**, **Variable Name for (Quote)**, **Variable Name for (Line)**, and/or **Library Function(s)** for the selected document. Clicking **Next Screen** takes you to the **Function Editor** page.*

6. **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.
7. **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*
8. Click one of the following actions:
    - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
    - **Apply:** saves changes and remains on the page.
    - **Update:** saves changes and returns to the Document List page.
    - **Back:** returns to the Document List page without saving changes.

Adding a Sub-Document (line level)

Documents can be created one at a time within a commerce process.

1. Enter a **Name**. This label distinguishes one document from another and is used as the page title on the commerce side whenever a user opens the document type. *This is a required field.*
2. Enter a unique **Variable Name** for your reference. *This is a required field.*
3. Enter a document **Description**.

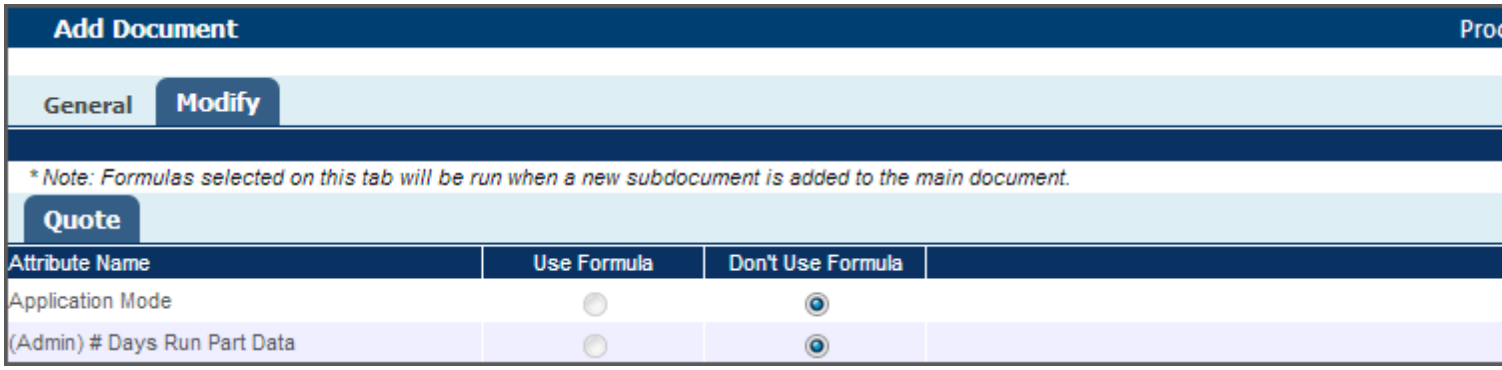


4. Select whether to **Run Scripts Once For All Line Items**.

5. **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.

6. **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*

7. Click the **Modify** tab to choose whether to **Use Formula** or **Don't Use Formula** for certain quote-level attributes. *Only quote level attributes exist on the line level Modify tab. For each of the quote level attribute the user can specify whether to use or not use formulas during a new line item addition.*



8. Click one of the following actions:
    - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
    - **Apply:** saves changes and remains on the page.
    - **Update:** saves changes and returns to the Document List page.
    - **Back:** returns to the Document List page without saving changes.

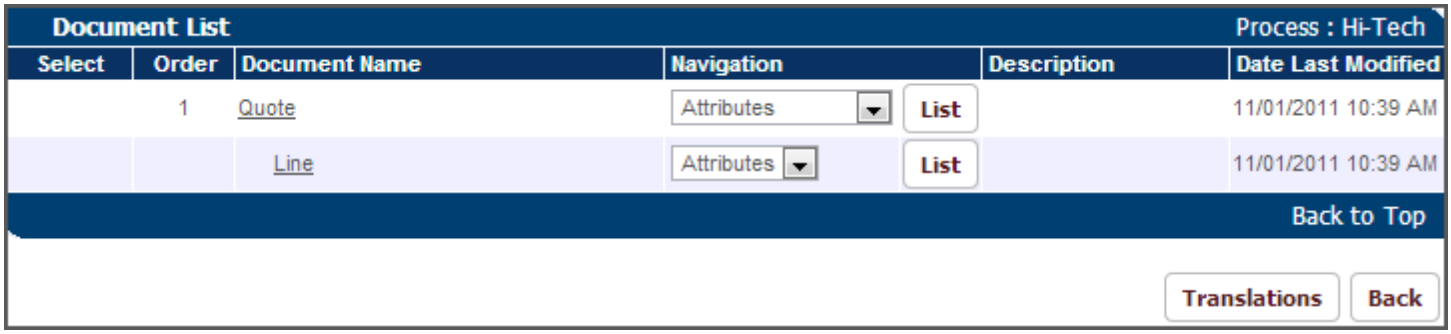


Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items.

Order of execution: Modify tab, Advanced Default- Before Formulas, Formulas, and Advanced Default- After Formulas.

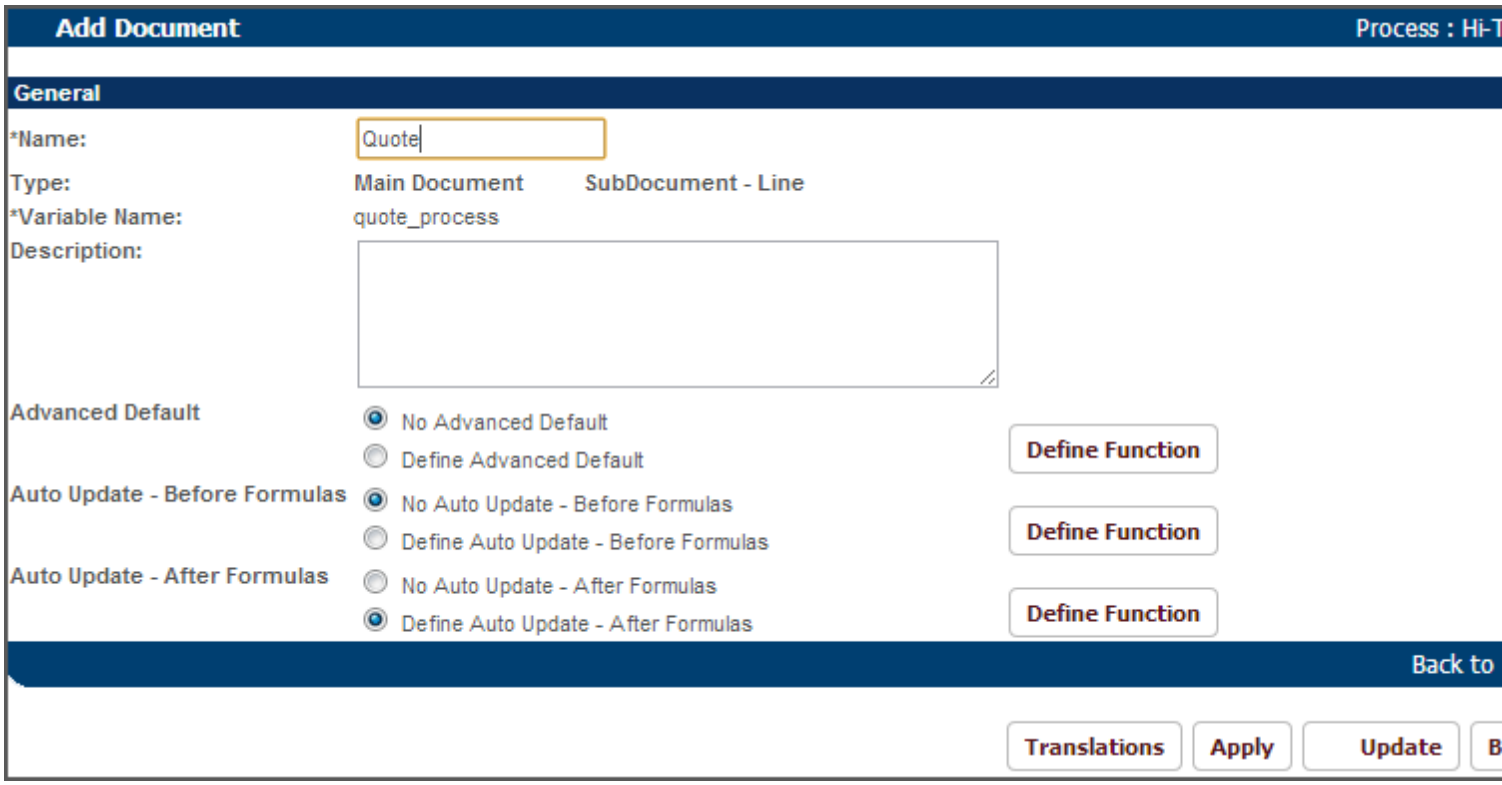
Editing a Main Document (quote level)

1. Click the name of the document you want to edit on the **Document List** page.



2. Edit the **Name** and/or **Description** displayed on the Add Document page.
3. Edit the following information, if necessary:
    - **Advanced Default:** Either select the **No Advanced Default** if you do not want to have an advanced default, or select **Define Advanced Default** and click **Define Function**. *This displays the **Select Attributes** window where you can select the **System Variable Name**, **Variable Name for (Quote)**, **Variable Name for (Line)**, and/or **Library Function (s)** for the selected document. Clicking **Next Screen** takes you to the **Function Editor** page.*
    - **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.

- **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*



4. Select one of the following options:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Apply:** saves changes and remains on the page.
   - **Update:** saves changes and returns to the Document List page.
   - **Back:** returns to the Document List page without saving changes.



Document names and descriptions can be edited at any time.

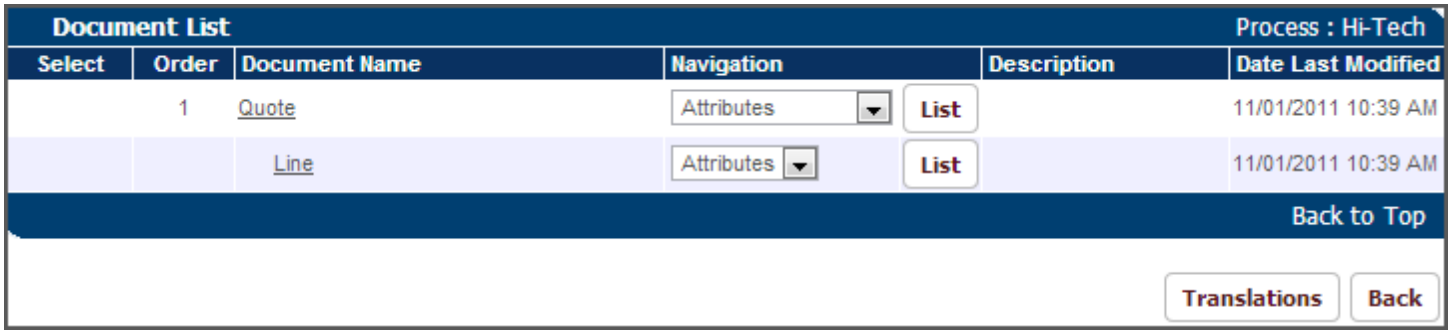


Quote level (main-document) formulas can be executed when users add a line item (sub-document) to

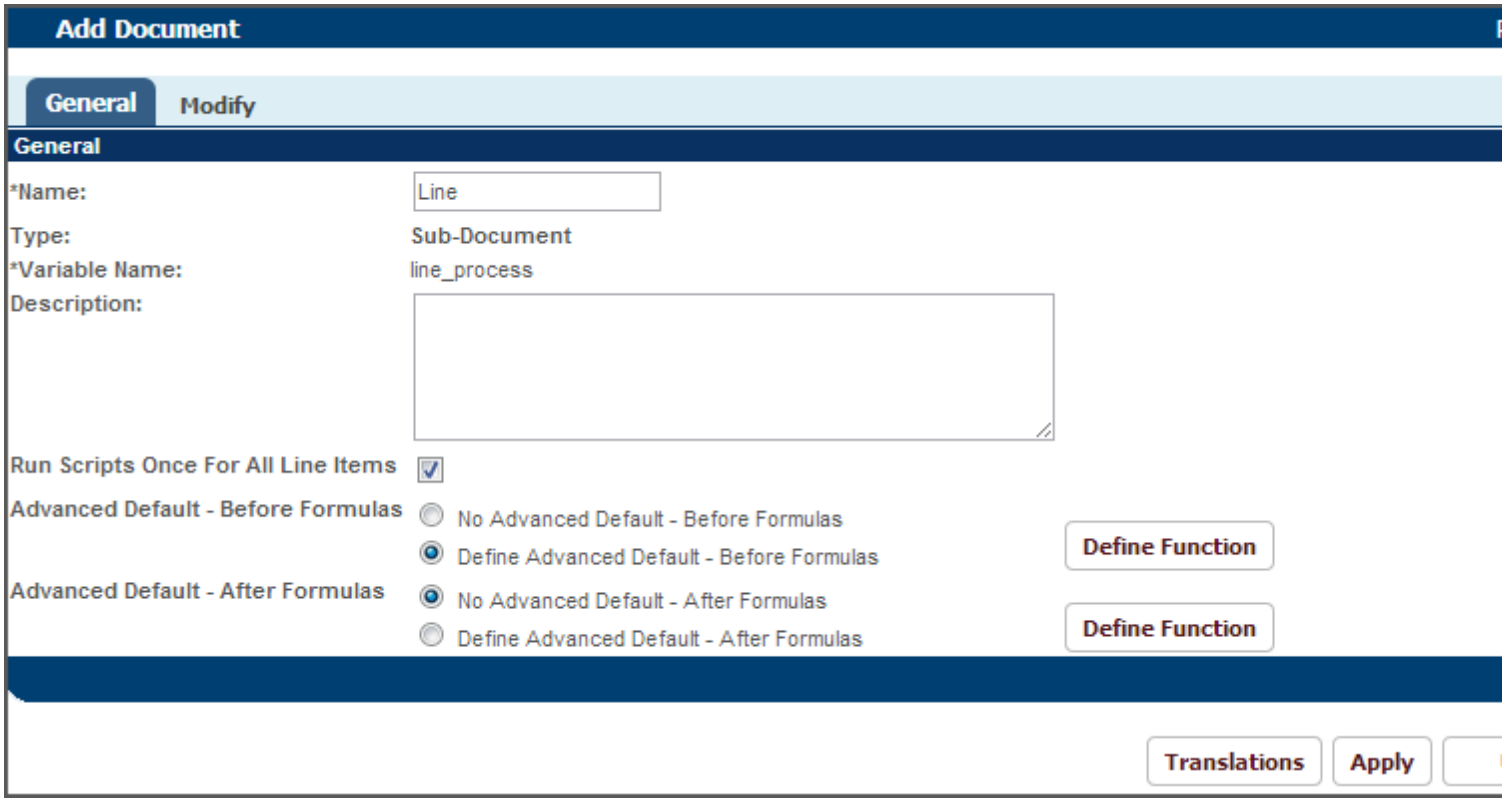
the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items.

Editing a Sub-Document (line level)

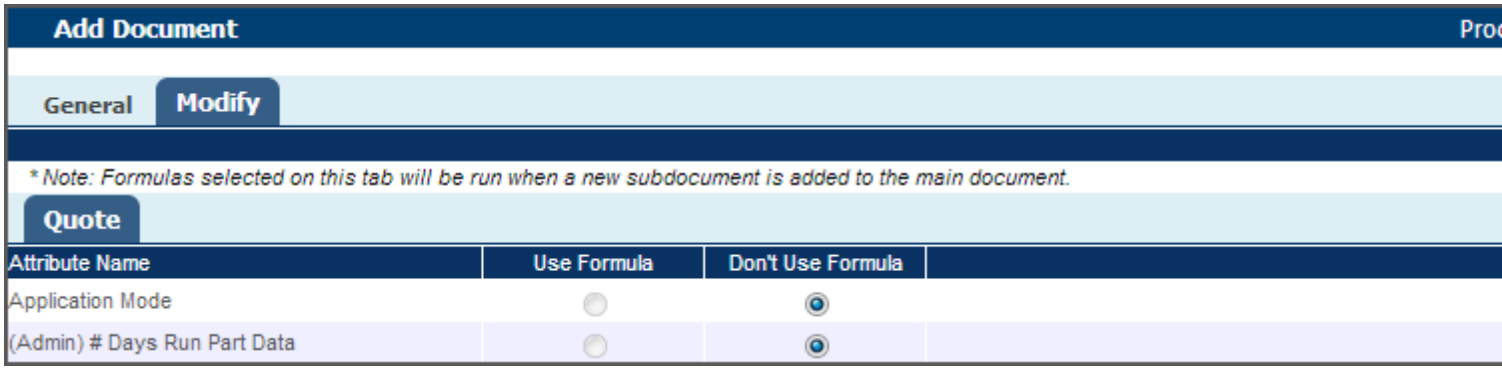
1. Click the name of the document you want to edit on the **Document List** page.



2. Edit the **Name** and/or **Description** displayed on the Add Document page.
3. Edit the following information, if necessary:
   - Select whether to **Run Scripts Once For All Line Items**.
   - **Auto Update- Before Formulas:** This allows functions to run before Formulas when the Commerce page performs an Auto-Update.
   - **Auto Update- After Formulas:** This allows functions to run after Formulas when the Commerce page performs an Auto-Update. *This was previously known as simply Auto-Update.*
   - Click the **Modify** tab to choose whether to **Use Formula** or **Don't Use Formula** for certain quote-level attributes.



4. Click the **Modify** tab to choose whether to **Use Formula** or **Don't Use Formula** for certain quote-level attributes. *Only quote level attributes exist on the line level Modify tab. For each of the quote level attribute the user can specify whether to use or not use formulas during a new line item addition.*



5. Select one of the following options:

   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Apply:** saves changes and remains on the page.
   - **Update:** saves changes and returns to the Document List page.
   - **Back:** returns to the Document List page without saving changes.



Document names and descriptions can be edited at any time.



Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items.

Order of execution: Modify tab, Advanced Default- Before Formulas, Formulas, and Advanced Default- After Formulas.

Deleting a Document

Use caution when deleting a commerce document because it is not possible to recover the deleted data. Upon a deletion, all associated views, attributes and actions within the document are also deleted.

1. Select the checkbox for the corresponding document you wish to delete on the Document List page.
2. Click **Delete**.

Ordering a Document

Ordering documents to determine how they appear in the **Commerce Manager**. Order numbers are auto-generated at the time of creating documents. As only one sub-document is associated with a main document, sub-documents don't require ordering.

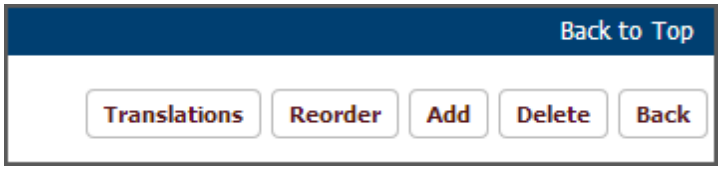
1. On the **Document List** page, click on the order number against the document and set it to any positive integer. *The order numbers of other main documents in the list will be automatically reordered.*
2. Click **Update**.



Changes to document order deletes current document mapping data.

Ordering Main & Sub-Documents

1. Navigate from: Document List > Quote or Line Attributes > List
2. Scroll to bottom of page and click **Reorder**.



3. Enter a new order number for the attributes you wish to reorder.

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*
        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*
    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



Edits are not visible until the process is deployed (or redeployed).

Create main documents as the primary documents in a set. Create sub-documents as detail pages that link up with main documents, that is, Line Item page for a RFQ.

There is an option "Run Once For All Line Items" next to the advanced default on sub-documents. Standard functionality of the Advanced Default is that it will run every time a line is added to a quote. Having this option enabled enhances performance.



Previously created variable names and previously set document types appear read-only because they are not editable.



Sub-documents and Main documents have a one-to-one relationship, meaning only one sub-document can be associated with a main document.

Main document Advanced Default will run when the transaction is first created and its purpose is to set main doc attribute default values. The purpose is to have one script that sets all of these attribute values as once, as opposed to pulling each of them separately. This also allows you to ease maintenance as all the defaults are set in a single script.

Sub-document (line item) Advanced Default will run whenever a line items is added to a quote. The purpose of writing this script is to set all of the line-item attribute values at once.



You cannot add, delete or update order numbers for documents in a deployed commerce process.

Certain areas of document administration are only visible for processes that have not been deployed. You cannot delete documents once a process is deployed.

User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

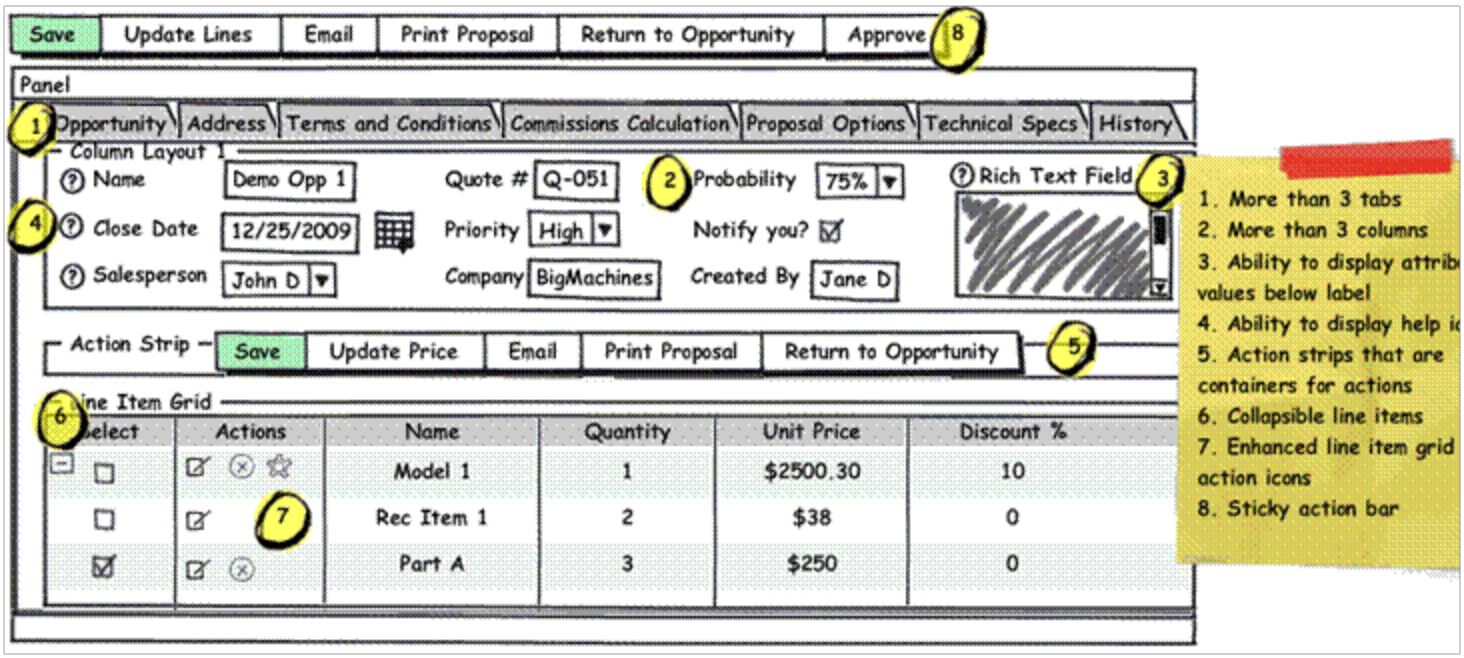
RELATED TOPICS

LAYOUT EDITOR OVERVIEW

OVERVIEW

You can customize and brand the user-side commerce UI in an easy and intuitive way thanks to drag & drop functionality, previews, grouping, and more. All of this is managed through the Layout Editor, which can be accessed from Quick Links, the Document List or by navigating to any main or sub-document attribute or action. The Layout Editor allows you to customize the look and feel of your commerce UI by using Outer Panels, Tabs, columns, Layouts, Line Item Grids, and the Commerce Stylesheet.

End-User Features:



Line Item Icon Actions

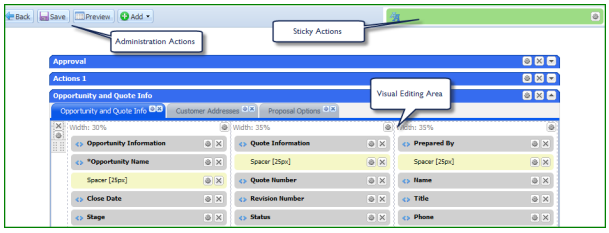
Action Icon	Description
Open Sub-Document	Allows user to view and edit the sub-document.
Reconfigure	Takes user back to a model configuration.
Add from Catalog	Adds items from the catalog.
Remove Line Item	Allows user to remove a corresponding line item if it isn't mandatory.
Collapse/Expand	Collapses and expands parent line items so that only the parent line item is visible.
Sticky Action Bar	Special action strip that follows the user down the transaction page as the user scrolls.

These actions are available to the user when they are active in the applicable Participant Profile.

Layout Editor

At the core of this new functionality is the new Layout Editor, which provides a graphical drag & drop feature for creating and managing document layouts. Users can have multiple columns and tabs, greater flexibility in laying out UI elements, and help text for attributes. *The Layout Editor is a fixed-width page that is divided into the following areas:*

Layout Editor Area	Description
Visual Editing Area	The Visual Editing Area allows the fullaccessuser to create and edit the components that make up the user-side layout of the commerce document. The layout editor provides a visual representation of what the layout is going to look like. *Drag & Drop technology allows fullaccessusers to move around layout elements including attributes, actions and tabs with a single click. Admins can change the order of the Outer Panels, the placement of visible Inputs and the order of the Commerce Tabs.*
Administration Actions Toolbar	These are actions that can be performed on the layout editor by the fullaccessuser.
Sticky Actions	The admin can manage which commerce actions to persist on the user-side. Sticky Actions are commerce actions that will always stay in the user's browser view , regardless of where the user scrolls on the page.



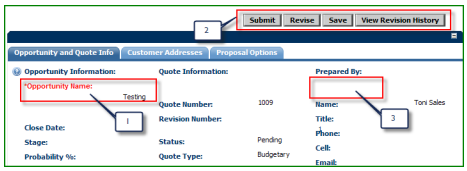
Layout Editor Elements

Element	Description
Panels	These are the outermost containers for layout elements. A commerce document can have multiple panels and the fullaccessuser can choose to show a label for a panel and make it collapsible, or to collapse by default.
Tabs	Tabs are what has replaced Groups, which contain column layouts, columns, attributes, action strips and actions.
Column Layouts	Each tab can have one or many column layouts.
Columns	Columns contain attributes, actions strips, and spacers.
Action Strips	Action Strips can contain one or more actions. The fullaccessuser can choose the width and alignment of the actions in an action strip.
Attributes	All attribute display properties are available in the Layout Editor and not the Attribute Editor. For example, rows and columns of a text area type attribute are availalbe in the layout.
Actions	These are placed in the action strips.
Spacers	Spacers have a default of 25px, though the fullaccessuser can set the height as necessary.
Line Item Grid	The fullaccessuser can define line item columns and map attributes to these columns. The Line Item Grid is available in the Main Document layout.

Administration Actions

Action	Description
	Upon clicking, the back button will take the fullaccessuser to the page they were previously on.
	Upon clicking, the layout will be saved. The asterisk is appended to the Save button to alert the fullaccessuser that changes have been made to the layout and they need to be saved.
	The fullaccessuser has the ability to preview the current layout without saving first.
	When the fullaccessuser clicks Add, a drop-down menu will open and allow the fullaccessuser to add Actions, Attributes or Layout Elements to the commerce document.

Input Elements



1. Attributes
2. Action Strip
3. Spacer

An Input defines a variety of field-level input objects. Each input has editable properties that are specific to their input type.

- Inputs are displayed as various color-coded "boxes"

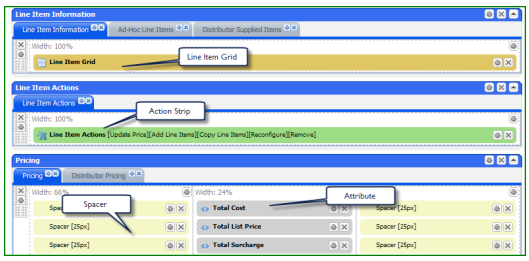
Color	Representation
Gray	Attributes
Green	Action Strips
Gold	Line Item Grid
Yellow	Spacers

- Each input has an associated icon, which is displayed as a box icon on the left side of each input.

Icon	Representation
	Action Strip
	Attribute
	Line Item Grid

- Inputs (regardless of type) can be dragged & dropped into any visible Input Column. The fullaccessuser can click and drag an input to a new column by clicking anywhere on the input box.

- Input property editing is available by clicking on the cog icon on the right side. The appropriate input editor dialog box will be opened.
- An input is removed from the layout by clicking on the 'x' icon on the right side of each input.



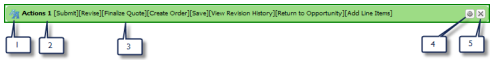
ADMINISTRATION

Action Strips

User-Side Action Strip:



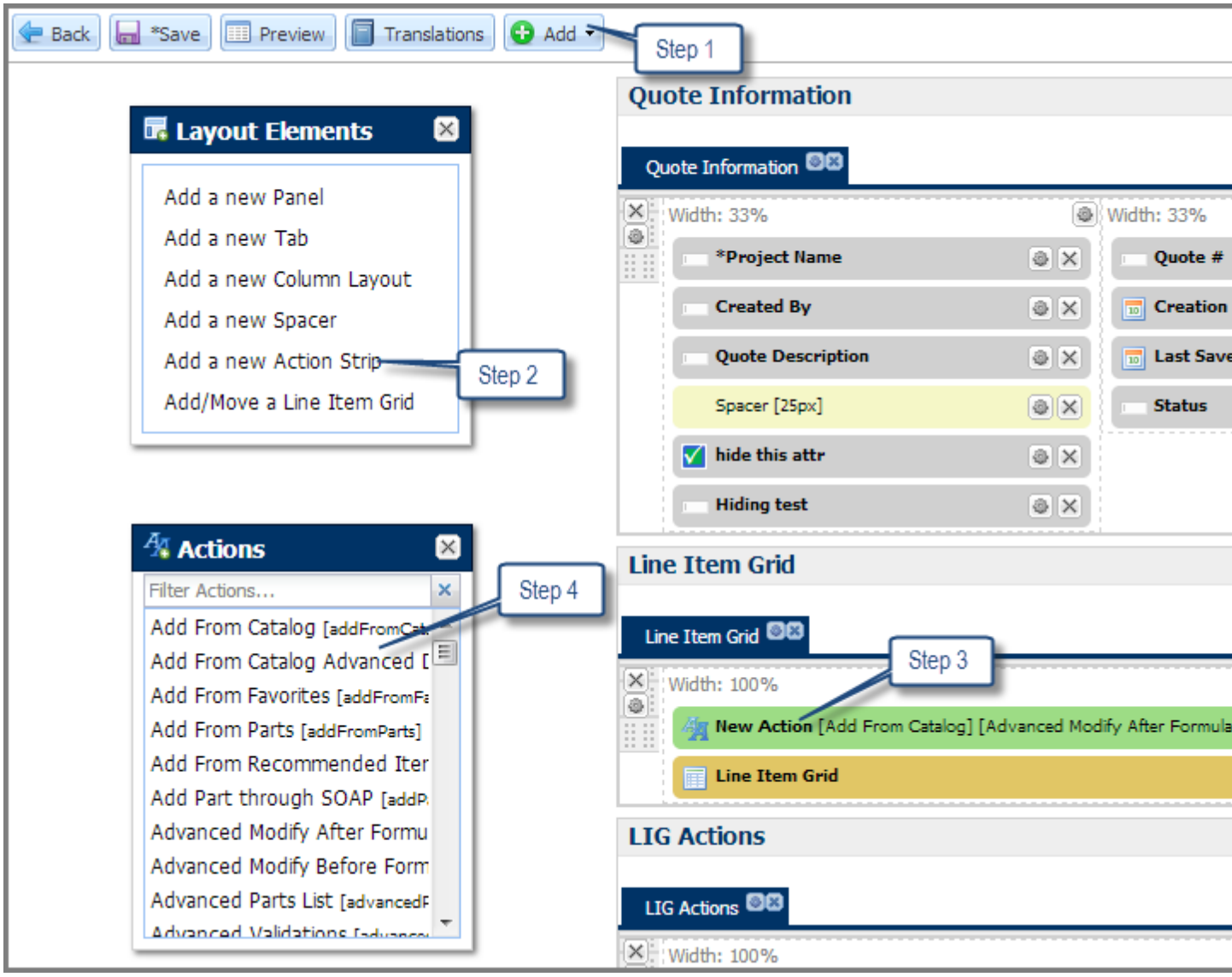
Action Strip in Layout Editor:



1. Action Strip Icon
2. Action Strip Label
3. List of Commerce Actions that belong to this Strip
4. Cog Icon to open Edit Properties Window
5. Delete Icon to remove the attribute from the column

Adding an Action Strip

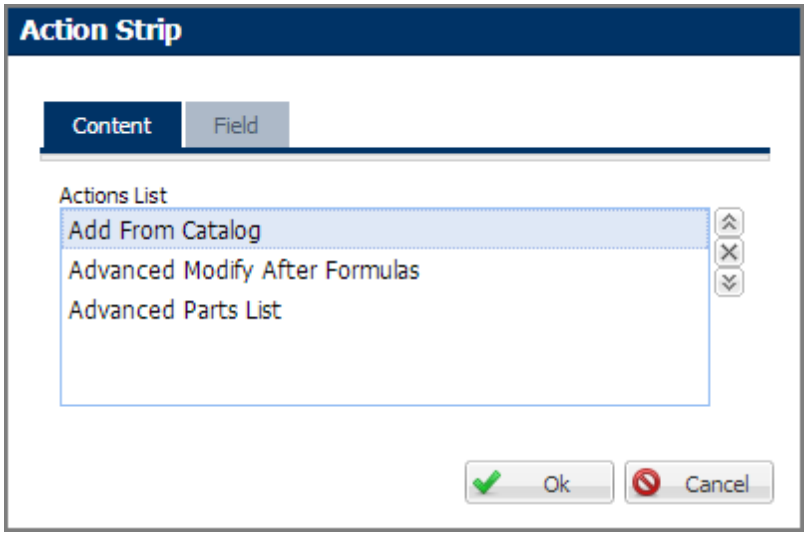
1. Click **Add** and select Layout Elements from the menu.
2. Choose **Add a new Action Strip** from the drop-down menu and drag & drop it into an Input Column. *The red dotted line represents where the action strip will be placed. The green check mark indicates that it can be dropped in that space.*



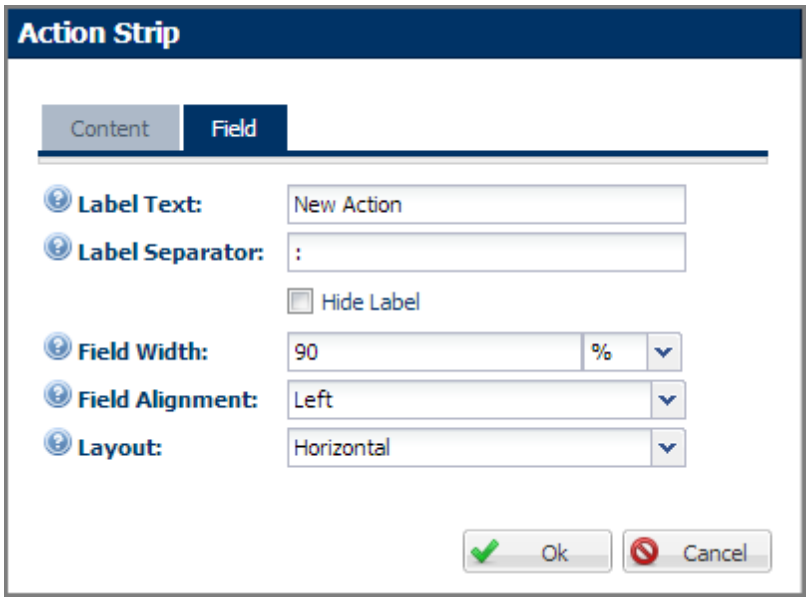
3. Repeat **Step 1** and click **Actions** to open the Actions dialog.
4. Drag & drop **Action** into your Action Strip.

Editing an Action Strip

1. Click on the ⚙ icon on the Action Strip. *The Action Strip Editor will opens in a pop-up window.*
2. Use scroll-bar in the **Content** section of the Action Strip Editor to view and select actions to order or remove.
3. After selecting an action, click the **Up** and **Down** arrows to re-order the actions or click the **X** to delete the action.
4. Click **OK** to **Save** and close the Action Strip Editor box or click **Cancel** to return to the Layout Editor without saving your changes.



5. Click the **Field** tab to make changes to the **Label**, **Width**, **Alignment** and **Layout**.
6. Enter the **Label Text** for the Action Strip. *If you wish to hide the label for the action strip, check the "Hide Label" box.*
7. Determine and input a label separator. *This is an optional character or text to append to the end of a label.*
8. Input the desired field width. *The field width can be defined in either percent or pixels. For best results, ensure that all columns in the layout add up to 100%.*

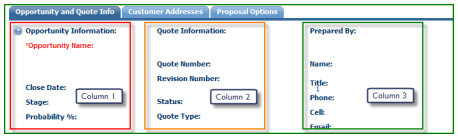


9. Determine the field alignment: **Left**, **Center** or **Right**. *This will determine how the buttons are aligned in the field.*
10. Determine if the actions should appear horizontally or vertically. *If horizontal, actions are laid side by side. If vertical, the actions are stacked on top of each other.*
11. Click **OK** to save changes and close the dialog box or click **Cancel** to close the dialog box without saving your changes.

Column Layouts

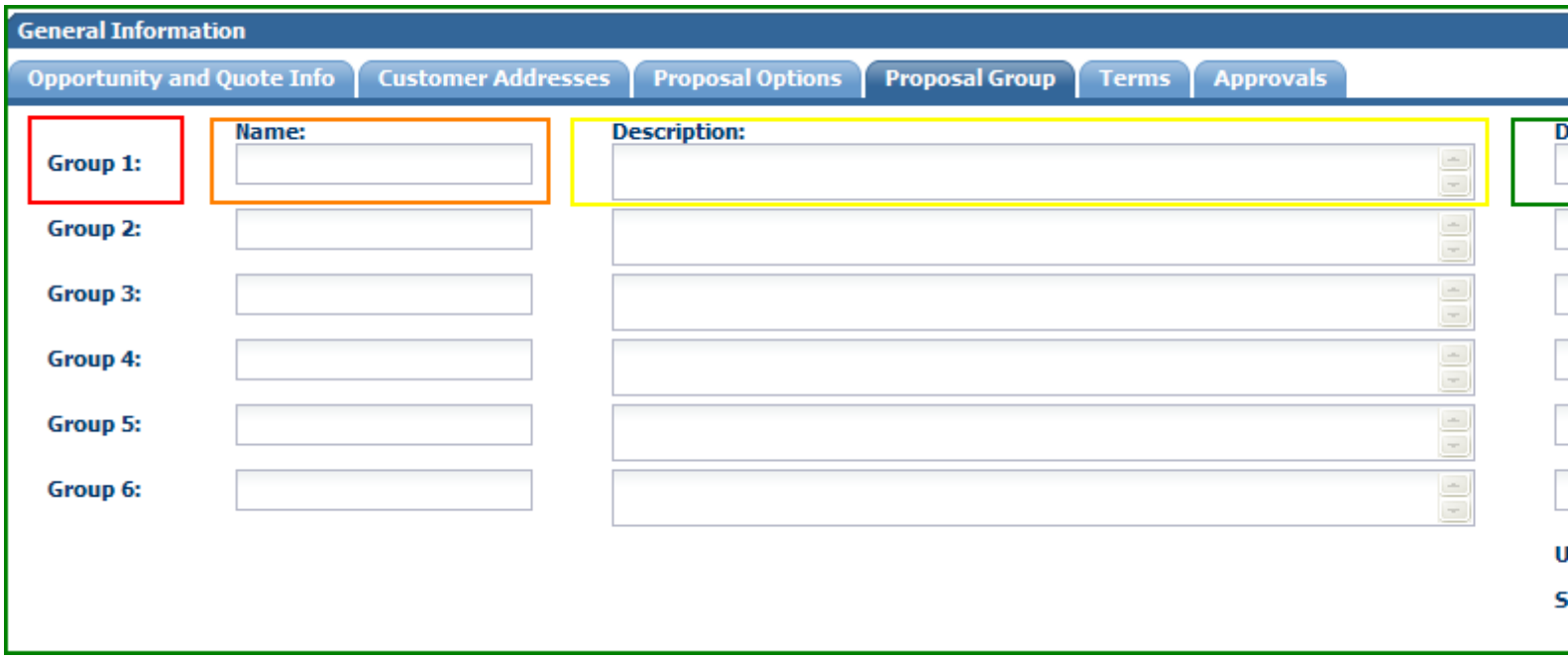
Column Layouts are represented as multi-column structures where each column contains Input Columns. A commerce tab is the only object that can contain columns layouts. Each layout has an independent column structure compared to neighboring layouts. There is no relationship between column layouts.

User-Side: 3 Different Input Columns inside a Commerce Tab



User-Side: Multiple Stacked Column Layouts

In order to mimic a data table within a tab, you will need to use multiple column layouts, and stack them. The image below is an example of this. On the first line, you will see five different columns highlighted in a different color. Each column is placed within a column layout and they have been stacked on top of one another. All the elements in groups 2-6 have each been placed in separate column layouts. **Example:** this layout is created with 6 column layouts and 5 columns within each layout.

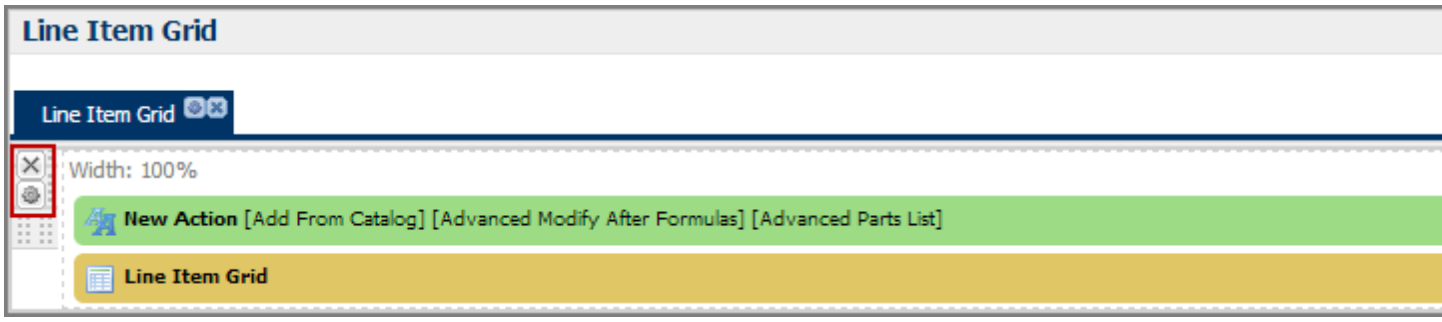


Icons

Icon	Description
	Opens the Panel dialog box.
	Removes the attribute from the column. **When you delete a Column Layout:** a warning message is displayed & the Column Layout is removed from the commerce tab.

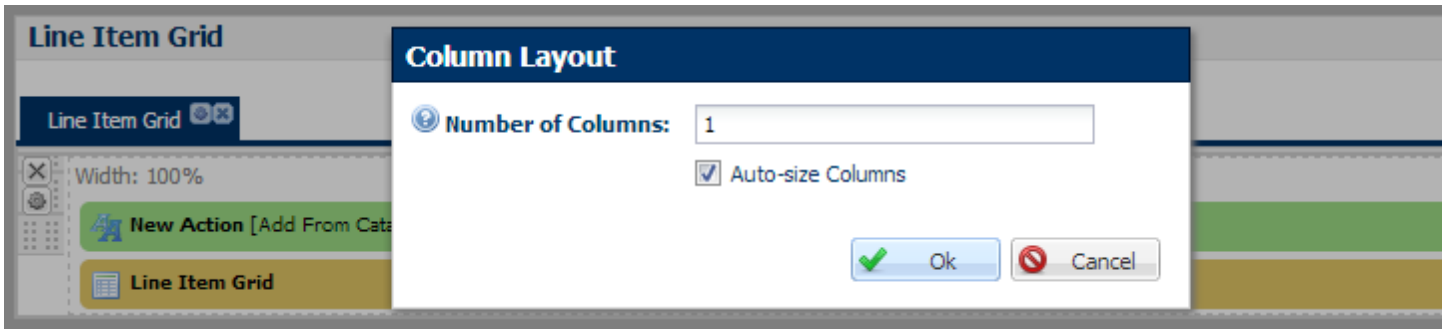
Adding a Column Layout

1. Click the icon.



2. Determine the number of columns that are required for the column layout. *There is a maximum of 20 columns allowed under one Commerce Tab.*

3. Select the **Auto-size Columns** box if the columns should be auto-sized. *Auto-Size will set the Column Layout to 100% and divide the size equally among the columns.*



4. Click **OK** to save the changes and close the dialog box or click **Cancel** to return to the Layout Editor without saving changes.

Horizontal scrolls are not provided by default. If the column width is bigger than the page width, then some fields may appear cut off on the transaction page. The admin can enable the horizontal scroll in the alternate stylesheet:

- **Fixed-width panels:** The following CSS will set the panels to always be 800px wide, so if the page is too big, space will be lost and if the page is too narrow, horizontal scroll bars will appear:

```
.panel{width: 800px}
```

- **Designate minimum width:** The following CSS is only supported on IE7 and FF3.5X:

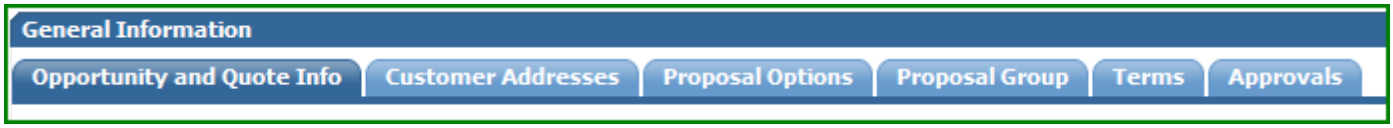
```
.panel { min-width:800px; /* IE doesn't understand min-width */ }

*html.panel{/*IE considers this legal whereas other browsers do not. Additionally, IE treats width
as other browsers treat min-width*/ width:900px}
```

Commerce Tabs

Commerce Tabs are displayed inside of an Outer Panel. There can be multiple tabs in a panel. Tabs can be mapped to hide parts, models or special product families. Tabs can also contain multiple column layouts within them.

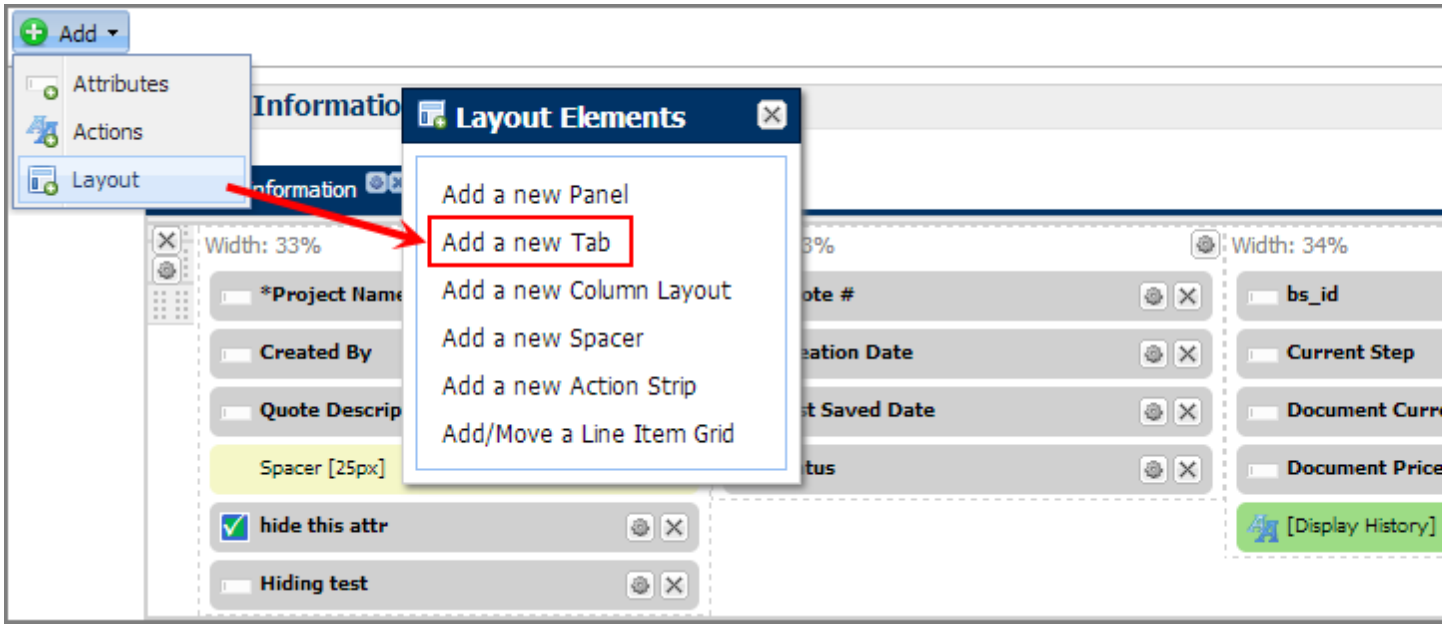
User-Side Commerce Tab:



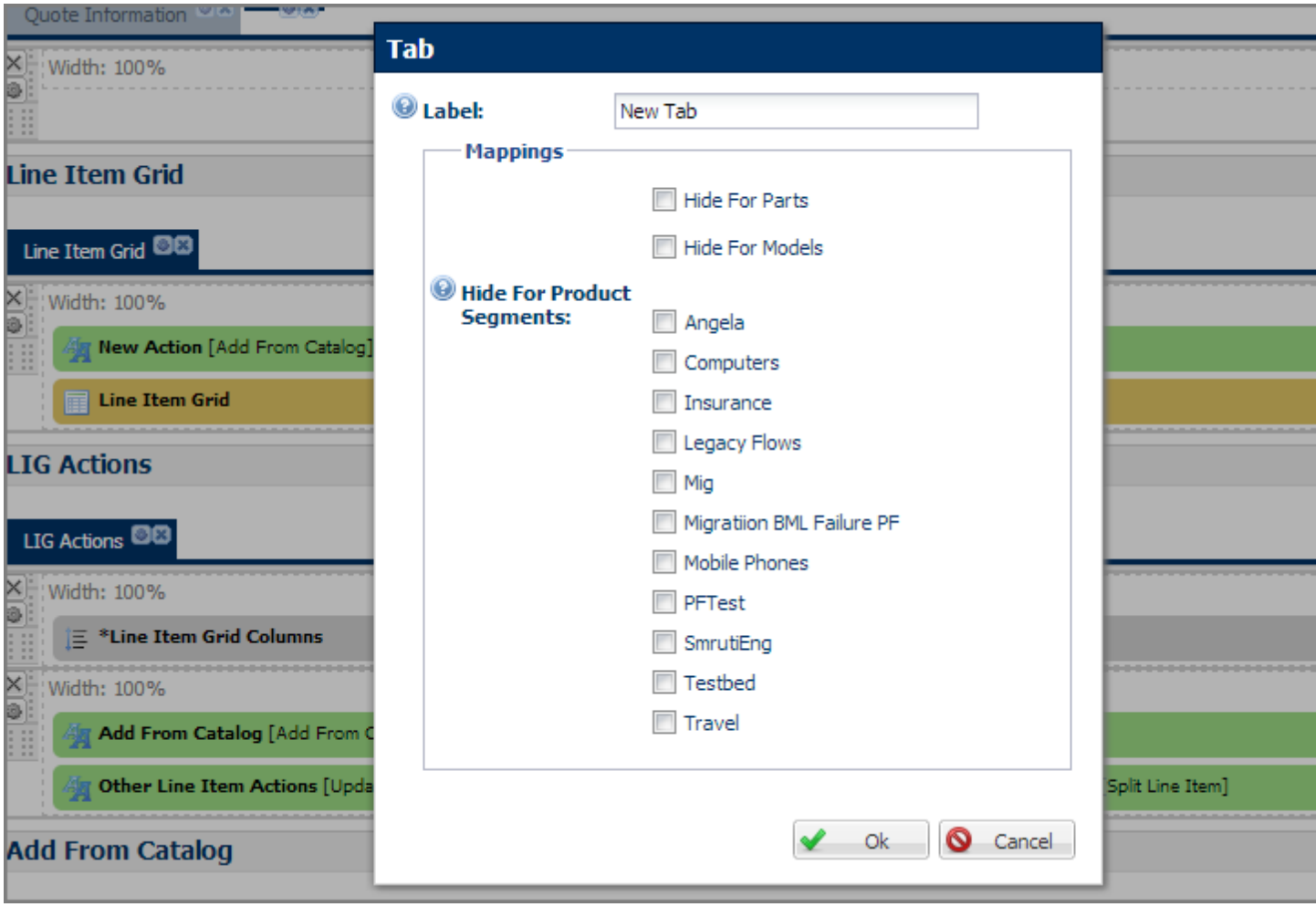
Icon	Description
	Opens the Panel dialog box.
	Removes the attribute from the column. **When you delete a Column Layout:** a warning message is displayed & the Column Layout is removed from the commerce tab.

Adding a Commerce Tab

1. Click **Add** in the Administration Actions toolbar.
2. Choose **Layout Element** from the drop-down menu.
3. Choose **Add a New Tab** from the menu.



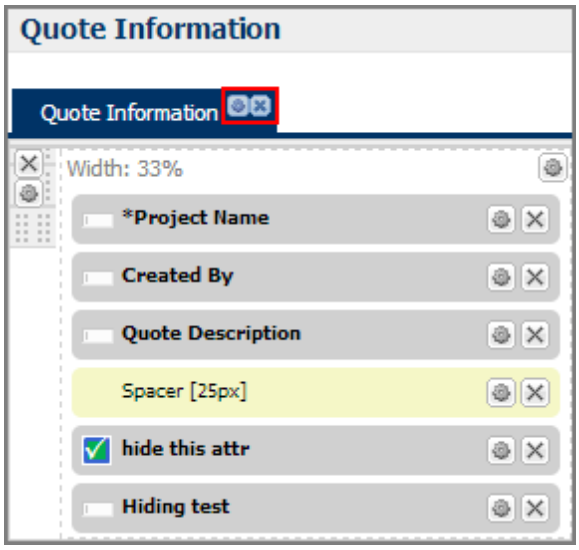
4. Drag & drop the tab into an Outer Panel. *The red dotted line represents where the panel will be placed. The green check mark indicates that its can be dropped in that space.*
5. Enter the **Label** text that should appear on the Tab.
6. Select the appropriate boxes if the tab should be hidden from any Parts, Models or special Product Segments.



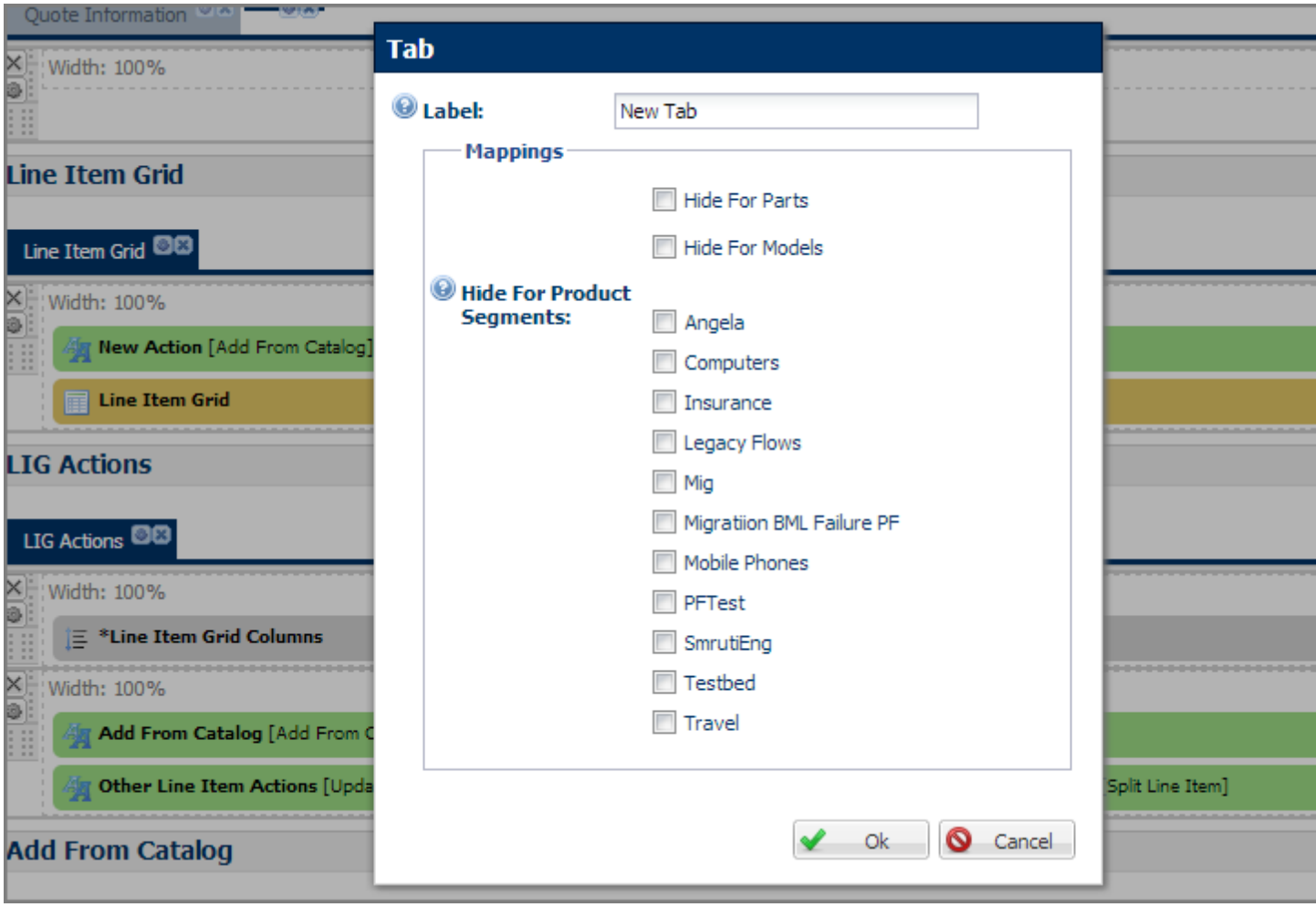
7. Click **OK** to save the changes and close the dialog box or click **Cancel** to return to the Layout Editor without saving changes.

---

1. Click the icon on the tab.



2. Enter the **Label** text that should appear on the Tab.
3. Select the appropriate checkbox if the tab should be hidden from any Parts, Models or special Product Segments.



4. Click **OK** to save the changes and close the dialog box or click **Cancel** to return to the Layout Editor without saving changes.

Moving a Commerce Tab

1. Select the tab to be moved.
2. Drag and drop the tab into its new position.
   - Displays if the move can be made.
   - Displays if the move cannot be made.

Input Attributes

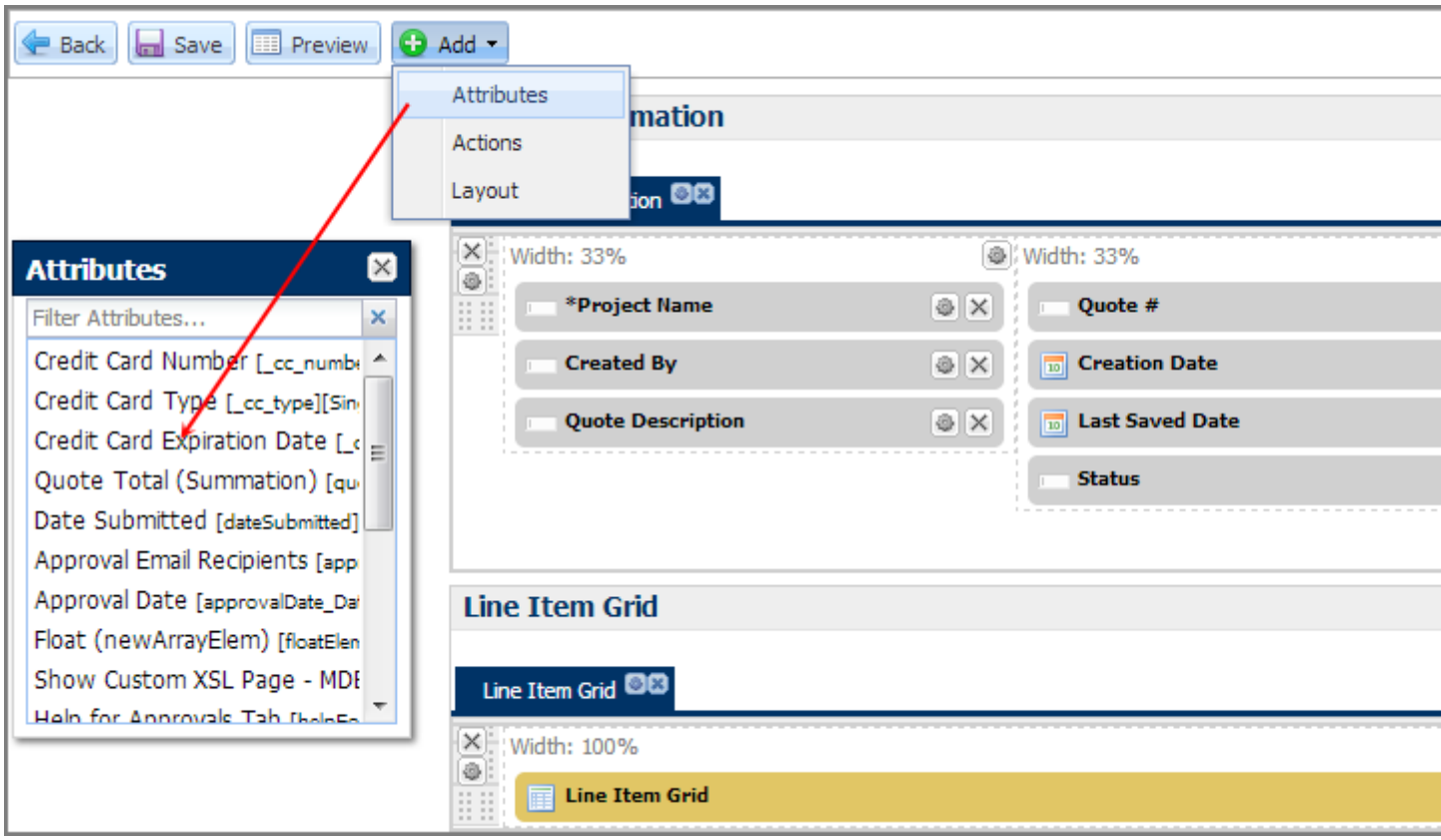
In the Visual Editor Area of the Layout Editor, an Attribute should be displayed with the following:



1. Attribute Name
2. Attribute Icon
3. Cog Icon to open Edit Properties Window
4. Delete Icon to remove the attribute from the column. When you delete an Input Attribute:

Adding an Attribute

1. Click **Add** in the Administration Actions toolbar.
2. Choose **Attributes** from the drop-down menu.

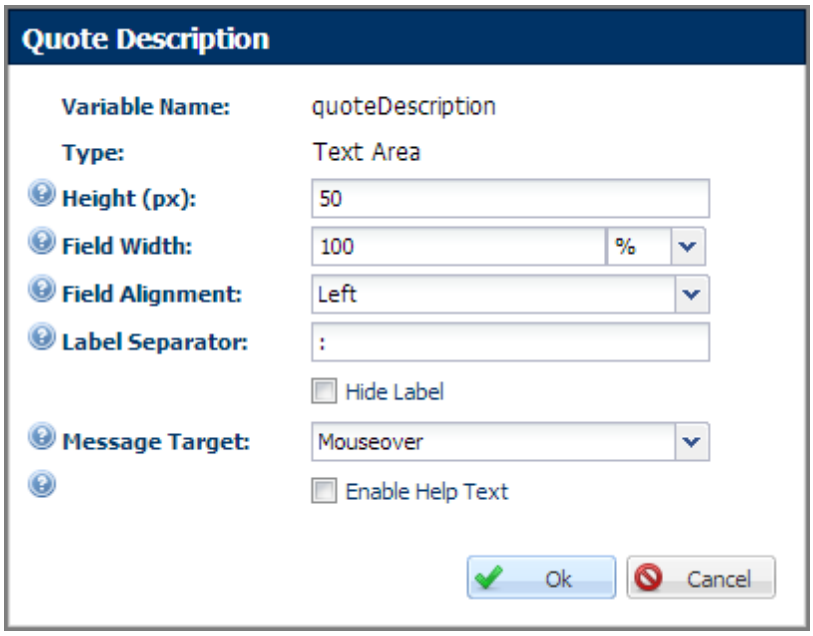


3. Scroll through Attributes and select one attribute.
4. Drag & drop Attribute into an Input Column. *The red dotted line represents where the attribute will be placed. The green check mark indicates that its can be dropped in that space.*

Editing an Attribute

When an fullaccessuser clicks on the icon of an Attribute field, an **Edit Properties** dialog box will pop-up. You will see the Attribute label displayed at the top right of the pop-up. You will also see that the Variable Name and Attribute Type are displayed. These fields cannot be edited.

1. Input the desired field width. *The field width can be defined in either percent or pixels. This is the %/pixel of the column that the field will occupy. If the value is greater than allowed, the column field will be cut off. The columns cannot add up to more than 100%.*

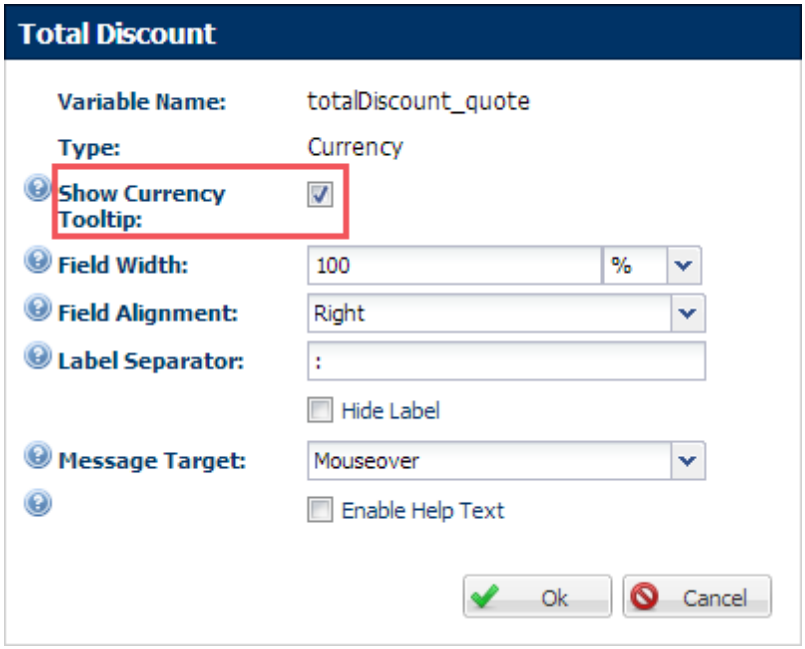


2. Determine the field alignment: **Left**, **Center** or **Right**.
3. Determine and input a label separator. *This is an optional character or text to append to the end of a label.*
4. Select the **Hide Label** box if you do not wish to display the Attribute Label.
5. Determine the location where an error text should be displayed:

Error Text Location	Description
Mouseover	Displays a quick tip when the user hovers over the field.
Right	Adds an error icon to the right of the field with a pop-up on hover.
Below	Adds a block beneath the field displaying the error text.

6. Select the **Enable Help Text**checkbox. *The Help Text is the description of the attribute.*
7. Click **OK** to save changes and close the dialog box or click **Cancel** to close the dialog box without saving your changes.

Currency Attribute Hover Tooltip: BigMachines 2013 Winter introduced enhanced decimal precision to include a hover tooltip that will appear if the stored precision value is different than the displayed precision and the numbers are not equivalent. This functionality is available only on currency attributes.

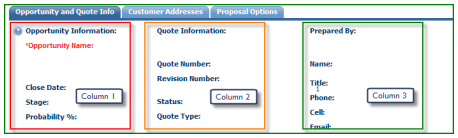


When you delete an Input Attribute: no warning message is given, the attribute becomes available in the **Add Attributes** dialog window, and the attribute is removed from the **Input Column**.

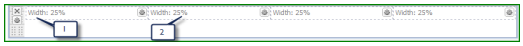
Input Columns

Input Columns are the columns that make up a Column Layout. They contain the commerce attributes, actions and the line items group.

User-Side Example: 3 Different Input Columns inside a Commerce Tab



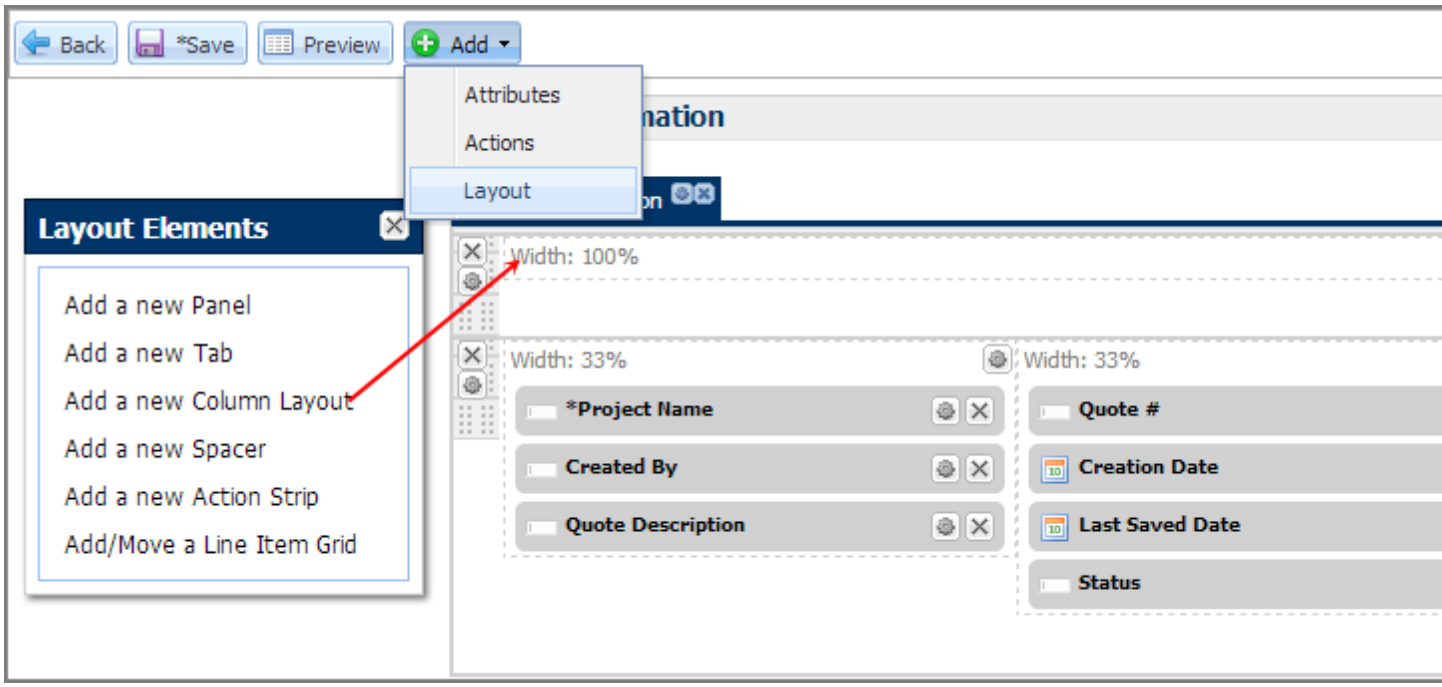
In the Visual Editing Area of the **Layout Editor**, an Input Column should be displayed with a:



1. Static Label displaying the column width.
2. icon to open the Panel dialog box.

Adding an Input Column

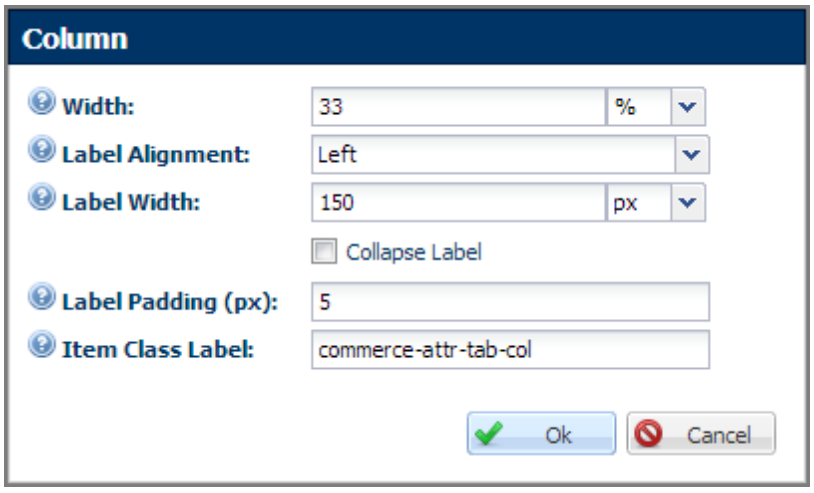
1. Click **Add** from the Administration Actions toolbar.



2. Select **Add a New Column Layout**.
3. Drag and drop new column into the desired location.

Editing an Input Column

1. Set the column width in pixels or percentages. *If the fullaccessuser chose to set-up the Column Layout and auto-size the Input Columns, then the width will be grayed out. For best results, it is recommended that all columns in the layout add up to 100%. Note: Columns cannot add to more than 100%.*



2. Determine the label alignment: **Left**, **Top** or **Right**.
3. Determine the label width in pixels. *The default width is 100 pixels. If the label is wider than specified, it will be cut off.*
4. Determine the label padding in pixels. *The default padding is 5 pixels.*
5. Enter an **Item Class Label**, if necessary. *The name of the custom CSS class to associate with this column. Using this class will allow you to style this column differently than the rest of this page. Let's say you wanted to change the color of the labels to green. You will need to add a custom CSS class to the alternate spreadsheet.* ***For example:***

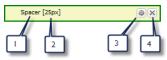
```
.my-class label.form-label
{
color: green.
}
```

6. Click **OK** to save changes and close the dialog box or click **Cancel** to close the dialog box without saving your changes.

> If "Collapse Label" is checked, this will remove the space reserved for the field labels.

Input Spacers

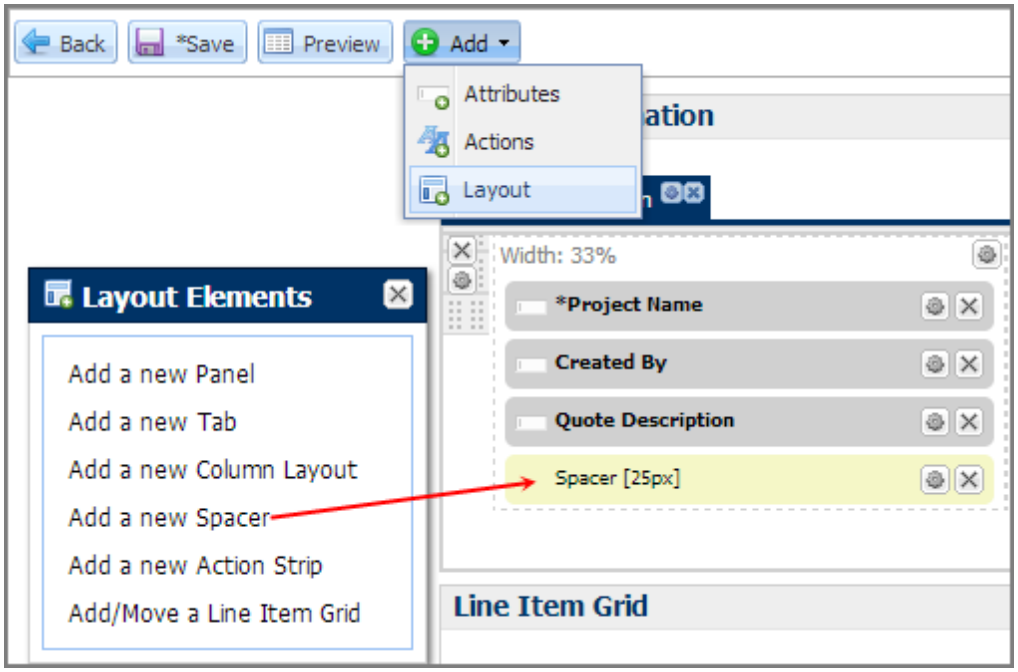
An Input Spacer is just a blank space with a defined height that is displayed vertically between other inputs. Spacers are displayed with a height that is defined by the fullaccessuser. In the Visual Editing Area of the Layout Editor, a Spacer should be displayed with the following:



1. Static Label: Spacer
2. A read-only display of the spacer's height value.
3. icon to open Edit Properties Window
4. Delete Icon to remove the attribute from the column.

Adding a Spacer

1. Click **Add** in the Administration Actions toolbar.
2. Choose **Layout Element** from the drop-down menu.

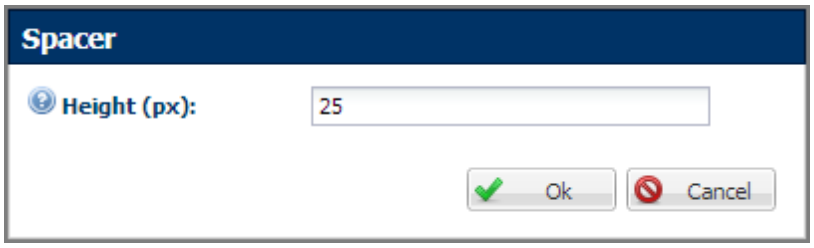


3. Choose **Add a new Spacer** from the menu.
4. Drag & drop the Spacer into an Input Column. *The red dotted line represents where the spacer will be placed. The green check mark indicates that its can be dropped in that space.*

Editing a Spacer Property

When an fullaccessuser clicks on the ⚙ icon of a Spacer field, an Edit Properties dialog box will pop-up. *The only property that can be edited on a Spacer is the height for the field.*

1. Determine the **Height (px)** in pixels.

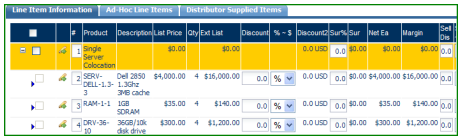


2. Click **OK** to save changes or click **Cancel** to return to the Layout Editor without saving the changes.

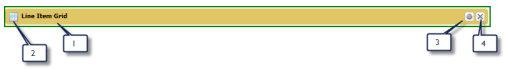
Line Item Grid

The Line Item Grid is a type of Input which represents the line item table in a commerce document.

User-Side Line Item Grid



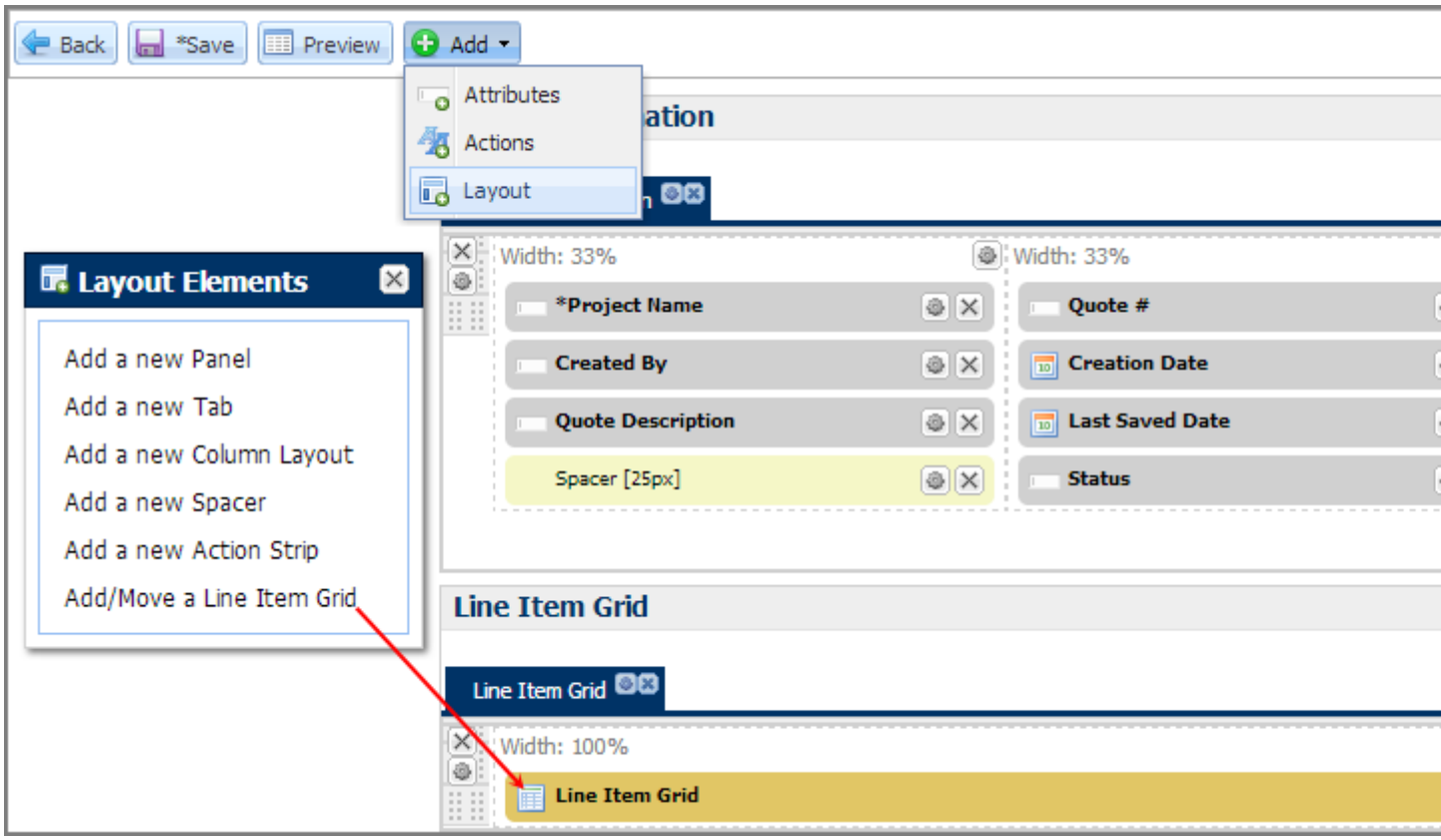
In the Visual Editor Area of the Layout Editor, a Line Item Grid should be displayed with the following:



1. Static Label: Line Item Grid
2. Line Item Grid Icon
3. Cog Icon to open Edit Properties Window
4. Delete Icon to remove the attribute from the column.

Adding a Line Item Grid

1. Click **Add** in the Administration Actions toolbar.
2. Choose **Layout** from the drop-down menu.



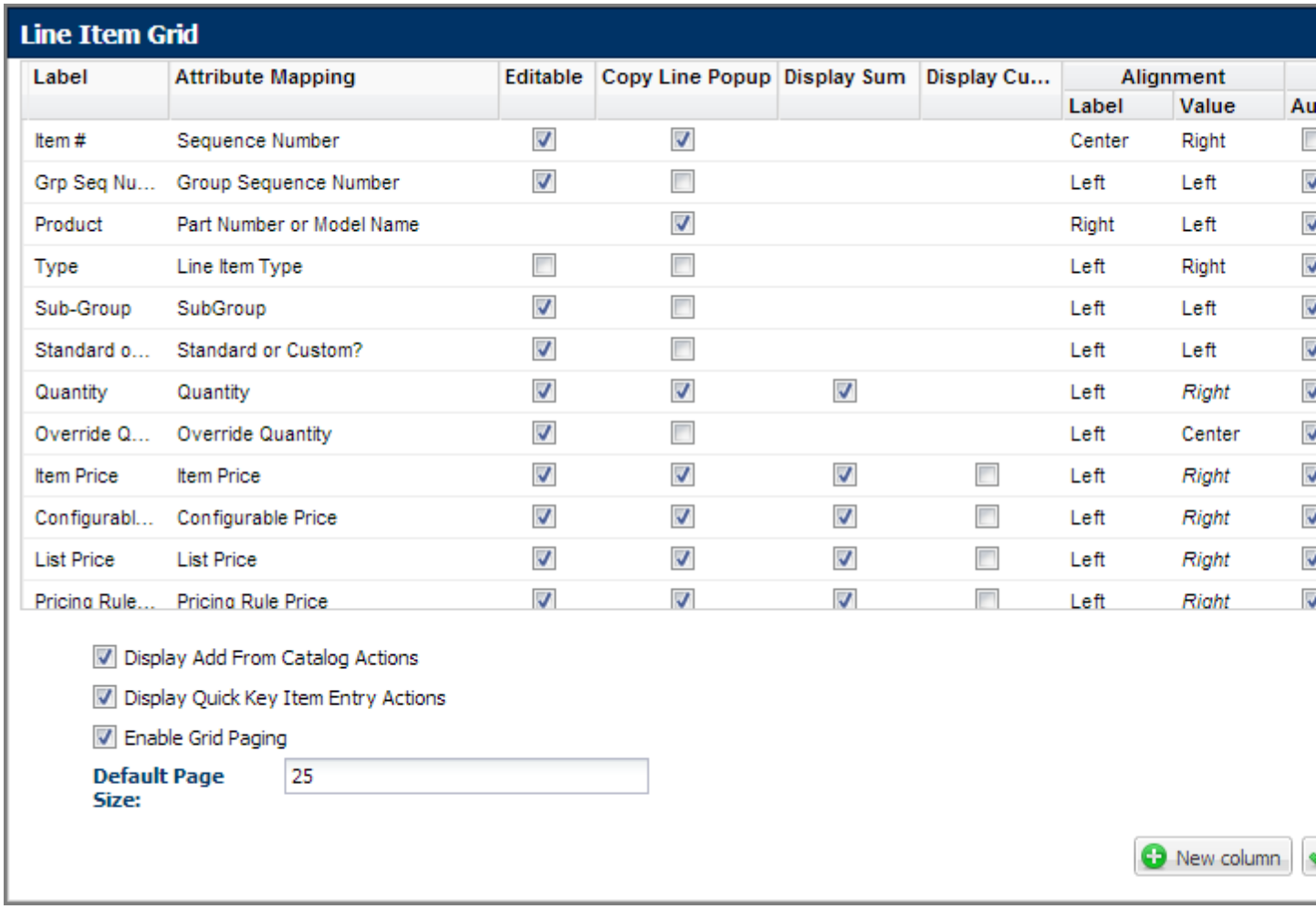
3. Choose **Add/Move a Line Item Grid** from the menu.

4. Drag & drop the **Line Item Grid** into an Input Column. *The red dotted line represents where the grid will be placed. The green check mark indicates that its can be dropped in that space.*

Editing a Line Item Grid

The columns table contains a vertical list of column objects. These columns are re-ordered through drag & drop functionality. The fullaccessuser will need to click Save to save the new order. To edit a line item grid, first click on the icon.

1. Click **New Column** if a new column needs to be added to the grid.
2. Double-click cell under **Label** and enter or edit the name of the column object.
3. Double-click the cell next to your new object beneath the header **Attribute Mapping**. *This will open the Column Mapping Editor Menu.*



4. Select the **Copy Line Popup** checkbox if the object should be editable and displayed as a pop-up for Copy Line Item.
5. Select the **Display Sum** checkbox to show the sum of all the values in the column to the user.

6. Determine the label and field alignments for each object: **Left**, **Right**, or **Center**. *Both the label and value default to the left.*

7. Determine the width of the column: **Auto-Width** or **Fixed-Width**.

Column Width	Description
Auto-Width	The width of this column is handled automatically by the grid. If auto is selected, the width value box is grayed out and disabled.
Fixed-Width	The width is fixed by the fullaccessuser. An integer text field contains the pixel value to use.

8. Click the **X** icon at the end of a column object to delete it.

9. Select **Display Add From Catalog Actions** checkbox if you wish to be able to Add Parts within Commerce.

10. Select **Display Quick Key Line Items** checkbox to enable Quick Key Line Items, if desired.

11. Select **Enable Grid Paging** to enable pagination of the line item grid. *On the transaction document, the user can control the number of lines visible by using the +/- icons to increment or decrement the number of line items on the page.*

12. Click **OK** to save changes or **Cancel** to close without saving changes.



When a Line Item Grid is in Read-Only mode, Boolean Attributes will display "True" or "False" (not as radio buttons)



When you delete a Line Item Grid: no warning message is given, the Line Item Grid becomes available to add to the layout again, and the Line Item Grid is removed from the current column.



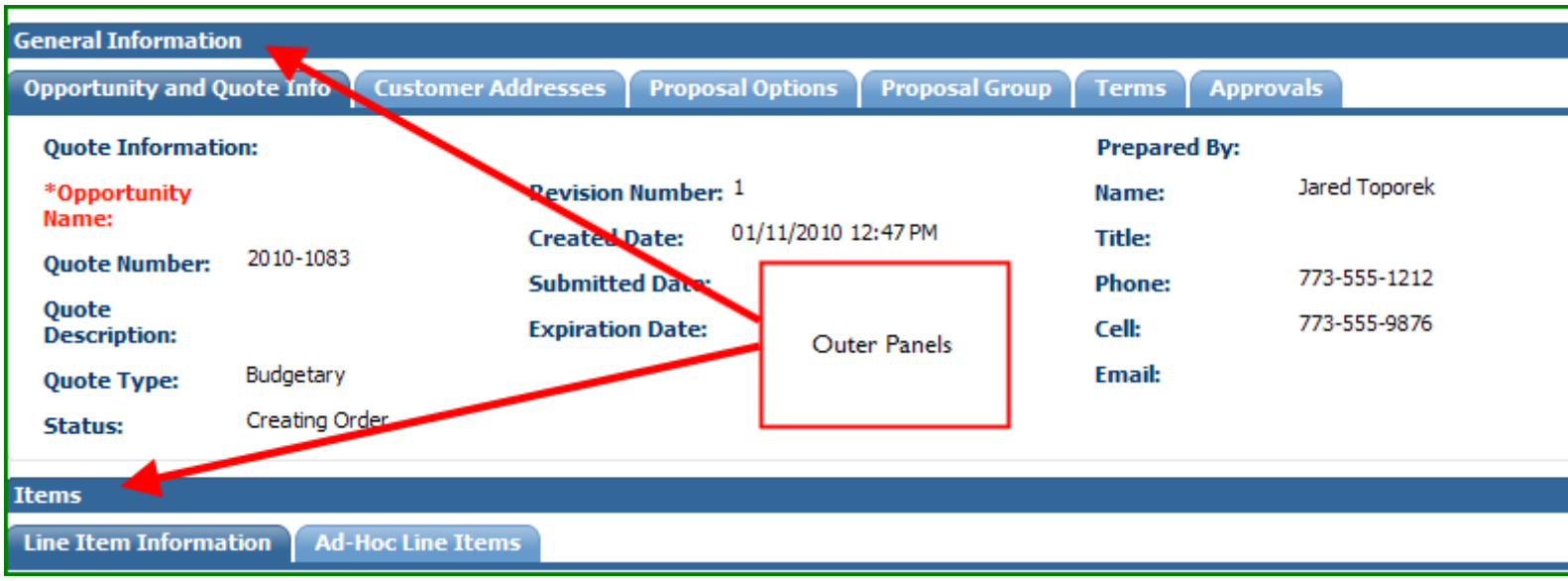
Do not display a Summation Attribute in the Line Item Grid. Never create a Summation Attribute at the line-item level.

The following Attributes CANNOT be displayed on the Line Item Grid:
-RTE Attributes
-HTML Attributes
-File Attachment Attributes
-History Attributes

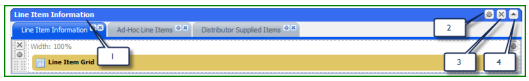
Outer Panels

Outer Panels are primarily used to group commerce tabs together. They provide the collapsing/expanding icon for the tabs as well as display a label that describes the tabs they contain. Layouts can have multiple panels. The fullaccessuser can choose to show or hide the label and whether to make the outer panel collapsible by default.

User-Side Outer Panels:



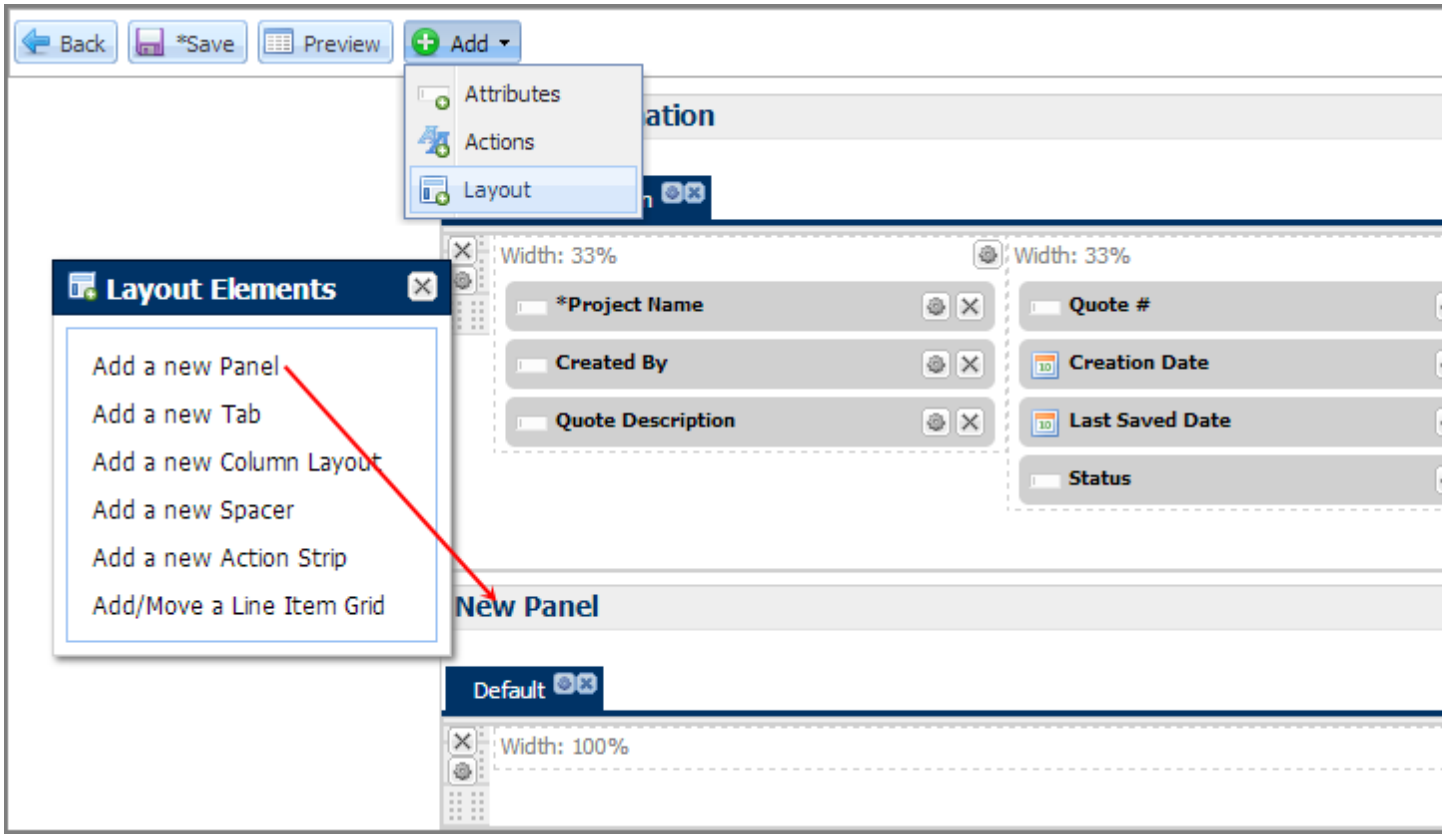
In the Visual Layout Editor, an Outer Panel should be displayed with:



1. Outer Panel Label
2. Cog Icon to open the Panel dialog box.
3. Delete Icon to remove the panel from the column.
4. Collapse/Expand Icon - To collapse a panel to make it easier to view the panels the user is working on.

Adding an Outer Panel

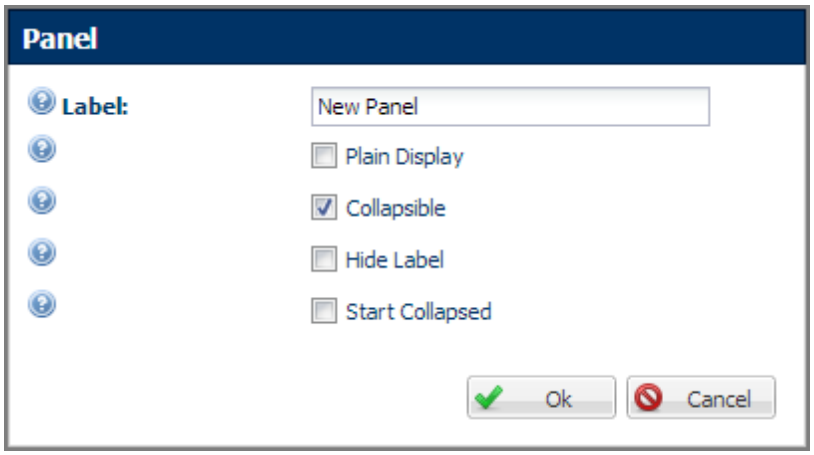
1. Click **Add** in the Administration Actions toolbar.
2. Choose **Layout** from the drop-down menu.
3. Choose **Add a New Panel** from the menu.



4. Drag & drop the panel into the Layout Editor. *The red dotted line represents where the panel will be placed. The green check mark indicates that its can be dropped in that space.*

5. Enter the **Label** text that should appear on the Panel.

6. Select the **Plain Display** checkbox to display the panel without labels or the ability to expand or collapse. *If this box is checked, the remaining options (Collapsible, Hide Label and Start Collapsed) will be grayed out. If you do not check the Plain Display box, you will be able to check the box labeled Collapsible. Selecting this option will allow users to collapse the panel. If this box is not checked, the Start Collapsed option will be grayed out.*



7. Select the **Hide Label**checkbox if you wish to display the panel without a label.
8. If you have chosen to make the panel collapsible, you can also choose to have the panel be collapsed when the page is initially loaded by selecting the **Start Collapsed**checkbox.
9. Click **OK** to save the changes and close the dialog box or **Cancel** to return to the Layout Editor without saving changes.

Clicking the edit icon will display the same Panel dialog box as the one that appears when you initially create the panel.

When you add an outer panel, one commerce tab is automatically created and labeled "Default".

You can drag & drop a panel once its been placed.

When you delete an Outer Panel:a warning message is displayed, and the outer panel is removed from the layout editor.

Sticky Actions

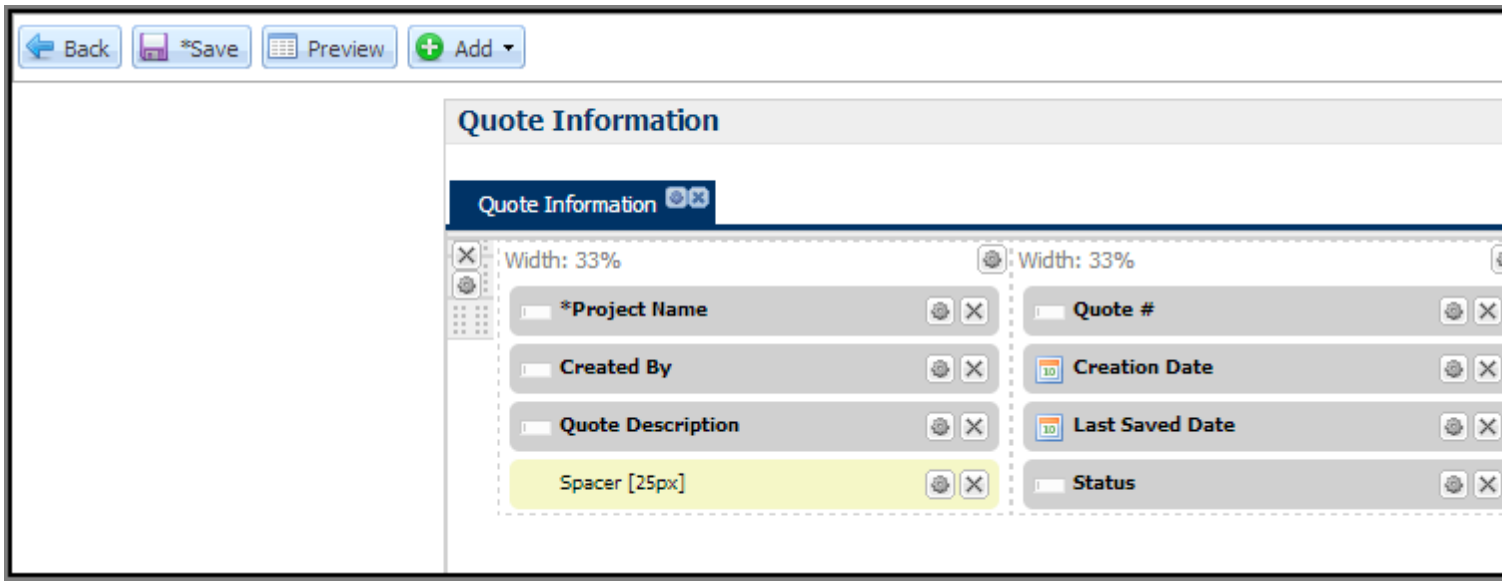
Sticky Actions are commerce actions that will always stay in the user's browser viewport, regardless of where the user scrolls on the page.

Sticky Action User-Side UI:

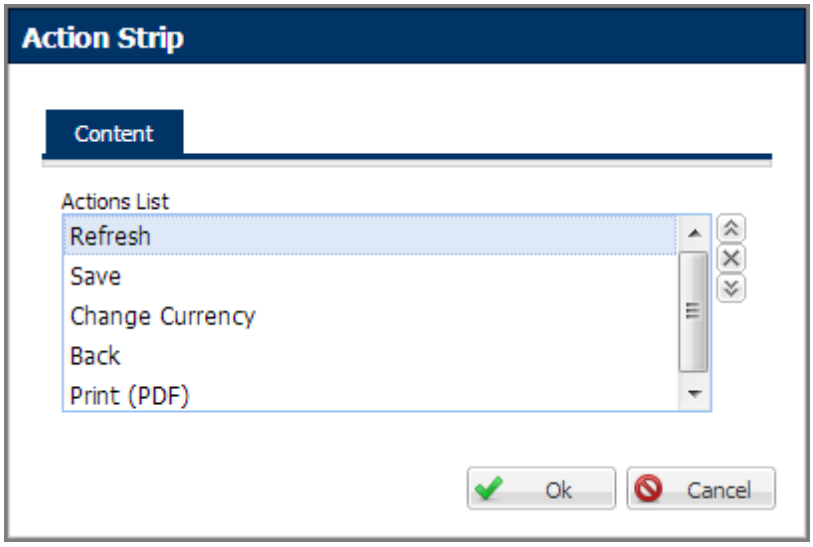


On the Layout Editor page, Sticky Actions are represented as an Action Strip to the right of the Administration Actions. The Sticky Action Strip should always be visible.

1. Click the icon in the Sticky Action Strip.



2. Use scroll-bar in the Content section of the Action Strip Editor to view and select actions to order or remove.
3. After selecting an action, click the **Up** and **Down** arrows to re-order the actions and the **X** to delete the action.



4. Click **OK** to Save and close the Action Strip Editor box or **Cancel** to return to the Layout Editor without saving your changes.

NOTES



Single-select menus and multi-select menus can be changed from one to another, but they first have to be removed from the layout and then changed in the attribute editor itself.

If labels intrude into the column space, either increase the label width or provide a space in the label text so that it can wrap.



The Main Document layout editor looks the same as the Sub-Document Layout Editor. The only exceptions are that in the sub-document layout editor, you cannot add attributes , you can only show line item attributes, and Add Action Strip only shows line item commerce actions.

Height of spacers, attributes and action strips within a column are consistent in the Layout Editor. All layout elements within a column are drawn with the same height on the Layout Editor for better usability. The fullaccessuser can observe the actual height of these elements on preview.

Numeric fields can only have the value field right-aligned in the line item grid.

Horizontal scrolls are not provided by default. If the column width is bigger than the page width, then some fields may appear to be cut off on the transaction page. The admin can enable the horizontal scroll by using the alternate stylesheet.



The fullaccessuser must be careful when using alternate stylesheets as it can affect the enhanced commerce UI adversely. Some things to look out for are resizing menu or text type attributes, placement of HTML attributes, etc. Also, the fullaccessuser should test any alternate CSS on all supported browsers since CSS behavior can vary greatly by browser.

Deleting a layout container like a panel, tab or column layout will delete all elements within the container as well.

You cannot mix pixels and percentages within a column layout.

HTML attributes that contained malformed HTML might break the UI. The admin is advised to check if an HTML attribute contains tags that are not enclosed in tags.

Including `prototype.js` in the header or footer of the transaction page may interfere with the functionality available to the end user, such as switching tabs, uploading files using the File Attachment

attribute, expand/collapse of line items, etc. Admins are discouraged from using prototype in any advanced customization on the transaction page.

RELATED TOPICS

COMMERCE ATTRIBUTES

OVERVIEW

Commerce document attributes are input fields for commerce documents. You can generate custom document attributes to develop a document, or you can use a combination of custom attributes and special attribute sets. The system provides default attribute sets for collecting data such as billing, shipping, credit card and pricing. All document attributes are administrated from the Attributes for Document page. After generating document attributes, you can set attribute defaults and icons, and map which products each document attribute applies to. In order for attributes to appear on documents, they must be included in document views. Custom document attributes are created one document at a time. It is not possible to copy attributes from one document to another.

Basic Attribute Categories:

- **Attributes:** Stand-alone input fields, which contain the transaction data. Attributes are available on the main document and sub-document.**Example:** a text field attribute or a menu attribute.
- **Attribute Sets:** These are a bunch of related attributes grouped together. Attribute sets are available both in the main document and sub-document. **Example:** a credit card attribute set or an address attribute set.

Attribute Types

Attribute Type	Usage	Editor Properties
File Attachment	Creates a file attachment field.	• **Exclude from PDF Merge:** This checkbox determines if the file uploaded to this attribute can be merged with the PDF proposal. • **Allowed Extensions:** This menu selection specifies that only selected extensions will be supported in the file attachment attribute. • **Available Tabs:** General, Document Views, Mapping, Groups
Text	Creates an input field for short strings.	• **Field Width:** This value determines the size of the text field. • **Maximum Field Length:** This value determines the maximum number of characters that can be input into the field. The valid range for this value is 1 through 255. • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Groups
Text Area	Creates a text area field for strings that are greater than 255 characters in length.	• **Number of Columns/ Number of Rows:** These fields allow you to specify the size of the text area field. • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Groups
Boolean	Creates a boolean field consisting of two radio buttons.	
Currency	Creates an input field and validations for monetary inputs.	• **Field Width:** This value determines the size of the text field. • **Maximum Field Length:** This value determines the maximum number of characters that can be input into the field. The valid range for this value is 1 through 255. • **Validation:** There are 5 validation options. Check Currency, Check Currency in Range, Check Currency for Maximum, Check Currency for Minimum, Check Currency for Non-negative. • **Maximum Value:** This field contains the maximum monetary value that can be input into the currency attribute. • **Minimum Value:** This field contains the minimum monetary value that can be input into the currency attribute. The minimum value must be less than the maximum value. • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Groups

Attribute Type	Usage	Editor Properties
Menu	Creates a menu field which may be multi-select or single-select.	• **Menu Type:** This field allows you to select the display type of the menu attribute. You can choose from Single Select Menu, multi-select Menu, Single Select radio buttons and multi-select check-boxes. • **Menu Population Box:** In the Menu Population section, create menu items one at a time. To add a menu item, enter the item name in the Menu Entry field, and click Add. To remove a menu item, high-light the item name, and click Delete. Menu items appear in the order they are listed in the Current Menu Entries list box. To reorder the list, highlight an item, and click the corresponding up or down button. You can also use the Entry Manager to manage your menu items. • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Groups
Integer	Creates an input field and val-idations for integer numbers.	• **Field Width:** This value determines the size of the field. • **Maximum Field Length:** This value determines the maximum number of characters that can be input into the field. The valid range for this value is 1 through 255. **Validation:** There are 4 validation options. Check Numeric, Check positive number, Check in range, Check positive number or zero. • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Groups
Float	Creates an input field and val-idations for floating point numbers.	• **Field Width:** This value determines the size of the field. • **Decimal Places:** This field determines the number of decimal places that will be supported. The maximum number of decimal places that can be supported is 9. • **Maximum Field Length:** This value determines the maximum number of characters that can be input into the field. The valid range for this value is 1 through 255. • **Validation:** There are 4 validation options. Check Numeric, Check positive number, Check in range, Check positive number or zero. • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Groups
Date	Creates an input field and val-idations for date inputs.	• **Field Width:** This value determines the size of the field. • **Includes Time:** This field determines the number of decimal places that will be supported. The max-imum number of decimal places that can be supported is 9. • **Maximum Field Length:** This value determines the maximum number of characters that can be input into the field. The valid range for this value is 1 through 255. • **Validation:** There are 3 validation options. None, Check for Future Date, Check for date in range. • **Start Date, End Date:** These fields need to be specified if the date in range validation is selected. *Validation can be set to accept all dates, all future dates or all dates within a range. When Check for Future Date is selected, the system ignores any specified start or end date.* • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Group
Summation	Creates a read-only field which dis-plays a result calculated from other attributes.	• **Format Output as Currency:** This checkbox determines if the summation attribute must have a formatted currency display. For e.g. if the advanced function of the summation attribute returns 556570, it will be displayed as $556,570. • **Evaluate on Save:** This checkbox determines if the summation attribute will be evaluated at the time of saving the document (performing a modify action). • **Define Function:** You can specify an advanced function to return a value for a summation attrib-ute. All calculations can be performed using the advanced function. The return value for a sum-mation attribute is a string. *If the "Format as Currency" option is enabled for the summation attribute, do not return a currency symbol in the script. The system will append the currency symbol auto-matically based on the currency of the transaction. If the "Format as Currency" option is not enabled, then you can return a currency symbol in the script.* • **Available Tabs:** General, Document Views, Mapping, Group Summation attributes are not edit-able.
History	Creates a text area field which	• **Number of Columns/ Number of Rows:** These fields allow you to specify the size of the text area

Attribute Type	Usage	Editor Properties
	always appends the new input along with the user's name and time.	field. • **Available Tabs:** General, Modify, Document Views, Mapping, Group
Sequence Number	Creates a field with the variable name `_sequence_number`. This attribute is created purely for ordering purposes and can be created only once for every sub-document in the system.	• **Field Width:** This value determines the size of the field. • **Order By:** This is a menu field that contains a list of all the attributes in the sub-document. *For best results, order by numeric, currency, date or text fields. Avoid using composite/ file attachment or read-only HTML fields.* • **Maximum Field Length:** This value determines the maximum number of characters that can be input into the field. The valid range for this value is 1 through 255. • **Validation:** There is only one validation for check positive.
Customer ID	This field is available only in the main documents of the process. Creates the special customer id field and its attached actions in order to integrate Accounts data.	• **Field Width:** This value determines the size of the field. • **Maximum Field Length:** This value determines the maximum number of characters that can be input into the field. The valid range for this value is 1 through 255. • **Available Tabs:** General, Default, Modify, Document Views, Mapping, Group
Composite	Composite attributes are usually part of system attributes that are delimited strings.	This attribute type is also unique in that you can control the display of composite attributes using XSL Views. Composite attributes are not editable.
Rich Text	Creates an attribute that can be populated by the end user on the commerce document. This attribute is available both on the main document and the sub-document.	• **Available Tabs:** General, Default, Document Views, Mapping, Group. The Document Views, Mapping, and Group tabs are similar to the tabs in other commerce attributes. *The special features of the General tab are:* • **Manage Templates:** Allows the user to create a new template or edit an existing template. The super user can choose to save a template in the Shared Template folder. This makes the template available for other rich text type attributes within the process. The user can also save the template in a folder created automatically for this attribute. This ensures that the template is available for this attribute only. The name of the folder is the same as the variable name of the attribute. • **Show Summary Text:** If Yes is chosen, then a summary of the attribute is shown to the end user. The number of characters shown depends on the Width, Height, and Show First values. If No is chosen, then a button is displayed. The user must click the button to launch the editor. • **Summary Text Options:** This allows the user to customize the way in which the rich text attribute is displayed to the end user. • **Height:** This gives the height of the displayed box. • **Width:** This determines the width of the displayed box. • **Show First:** This determines the number of characters that are shown to the end user. To see the entire content, the user must double click the text box. *The special feature of the Default tab is:* • **Select a Template:** If this option is selected, the attribute is pre-populated by default with the chosen template. *Rich Text Attributes are not supported in the Commerce XML and cannot be accessed through SOAP, Web Methods or Bulk Upload/Download.*
Read-Only Text or HTML	Creates a read-only field for text or HTML.	

System Attributes

Commerce system attributes are used in rules as inputs to determine certain behavior. *Besides the attributes listed in the table below, other user attributes, including user phone, user email, user billing city etc (from the user administration page) are also available.*

System Variable Name	Description
_system_date	Returns the current date. Data type is string. **Example:** 6/20/2011 15:48:32
_system_current_document_number	Returns the current document number. Data type is string. **Example:** a line item document could have a value as "3".
_system_current_document_language	Returns the current document language. All documents in a transaction are in the same language. **Example:** "en" stands for English - United States.
_system_current_document_currency_pref	Returns the current document currency preference. Data type is string. All documents in a transaction have the same currency preference. **Example:** Currency preference code of "USD", which would stand for United States Dollars.
_system_base_currency	Returns the base currency of the host company. Data type is string. **Example:** Currency preference code of "USD", which would stand for United States Dollars.
_system_currency_conv_rate	Returns the currency conversion rate for the host company. Data type is float. **Example:** 2.7
_system_company_name	Returns the name of the host company. Data type is string.
_system_step_var	Returns the variable name of the current step that the transaction is in. Data type is string. **Example:** "test_step1".
_system_buyside_id	Returns the unique numeric buyside ID of the current transaction. Data type is string. **Example:** "901234"
_system_add_from_catalog_action	Returns the add from catalog action name. Data type is string. **Example:** "Add Line Items".
_system_transaction_counter_value	Returns the transaction counter value. Data type is Integer. You can map an attribute to this system defined attribute in the Modify tab.
_system_user_session_id	Returns the unique session ID of the user who is currently logged into the application. Data type is String.
_system_user_partner_session_id	Returns the partner session ID of the current user. Data type is String.
_system_user_name	Returns the current user's name (first name and last name). Data type is string. **Example:** John Smith.
_system_user_login	Returns the current user's login. Data type is string. **Example:** jsmith
_system_user_email	Returns the current user's email address. Data type is string. **Example:** abc@xyz.com
_system_user_currency_pref	Returns the current user's currency preference. Data type is string. **Example:** usd
_system_user_number_format	Returns the number format preference of the current user. Data type is string. **Example:** ####.##
_system_user_	Returns the date format preference of the current user. Data type is string. **Example:** MM/dd/yyyy h:mm a

System Variable Name	Description
date_pref	
_system_user_time_zone	Returns the time zone of the current user. Data type is string. **Example:** US/Eastern
_system_user_units	Returns the units of the current user. Data type is string. **Example:** Metric
_system_user_groups	Returns a plus delimited string of all the groups that the current user belongs to. Data type is string. **Example:** marketing+sales_rep+order_fixer+vpsales
_system_user_type	Returns the system defined type of the current user.
_system_user_is_notify_email	Returns a boolean value for the current user's notification preference. Data type is boolean. **Example:** true
_system_modify_order	This system attribute is always initialized to a constant value (-1) for the original line items . For copied line items, it returns the value as a constant value (1). In case of a successful **Reconfigure** action, the value changes to the constant (2). In case of existing quotes, adding the **Copy Line Item** action, initializes the attribute to (-1) for all line items. *This variable is typically used in scripts, and the values for the variable is stored after the action is performed.*
_system_selected_document_number	Returns the document number of the selected line item within the main document. *This helps identify the line item selected for all actions.* Selecting multiple line items returns the document numbers of the line items in the form of a tilde delimited string. This is typically used in scripts and the values of the variable are not stored after the action is performed. This variable is populated in all commerce main document actions. The fullaccessuser is able to access a tilde-delimited list of selected documents in the modify tab, advanced modification and advanced validation functions of main document actions. *This tilde-delimited list of document numbers cannot be returned from the Advanced Modification.*
_reconfiguration_date	Returns the reconfiguration date of the model. This is automatically populated each time the user runs reconfiguration. **Example:** in case of a non-configurable line item, **Parts**, the system returns a constant value (-1). The variable is initially initialized to January 1, 1970 for parts and models, yet to be re-configured. *In case reconfiguration fails, this date remains unchanged.*
_last_modified_date	Returns the last modified date of the sub-document; string "1" for the 1st modify and "2" for the second modify. This variable is initially initialized to January 1, 1970 for sub-documents; once the sub-document is modified, this is reset to the new modify date.
_is_recommended_price	Returns a boolean value to help identify whether the line item has a recommended price. ***True*** *indicates that the price is derived from the advanced pricing in the recommended items rule.* ***False*** *indicates that the price is not derived from the advanced pricing in the recommended items rule.*

Quote Attributes

Variable Name	Label	Description
_quote_process_document_number	Document Number	Unique doc number to be used as a document identifier in a commerce process.
_quote_process_price_book_name	Price Book Name	Name of price book used in transaction.
_quote_process_price_book_desc	Price Book Description	Description of price book used in transaction.
_quote_process_price_book_custom_fields	Price Book Custom Fields	Custom fields used in the transaction price book.
_quote_process_price_book_custom_field_text	Price Book Custom Field Text	Custom field text used in the transaction price book.
_quote_process_partner_price_book_id	Partner Price Book ID	Partner Price Book ID of transaction price book.

Variable Name	Label	Description
_quote_process_customer_id	Account ID	Standard attribute that allows user to look up contacts and accounts. *Required.*
_quote_process_billTo_	Bill To First Name	Bill to address and contact fields.
_quote_process_shipTo_	Ship To First Name	Ship to address and contact fields.
_quote_process_submitted_by_submit_quote	Submitted by [Submit]	Login of Submitter. Only populated after corresponding submit action is performed.
_quote_process_document_link	Document Link	Link to the transaction that needs approval. *This is only populated in advanced notification email rules.*



All the above Quote Attributes utilize a String data type.

Line Attributes

Variable Name	Label	Data Type	Description
_partner_entity_id	Partner Entity ID	String	
_document_number	Document Number	String	Unique document number to be used as an identifier in a commerce process
_parent_doc_number	Parent Document Number	String	Used to identify the parent document number, if any, for this sub-document.
_parent_line_item	Parent Line Item	String	Used to identify the name of the parent line item, if any, for this sub-document.
_sequence_number	Line #	Integer	Sequence number for the document.
_price_quantity	Quantity	Integer	Quantity of the line item.
_price_override_quantity	Override Quantity	Boolean	Select if you need to override the original quantity.
_price_list_price_each	Unit Price	Numeric (currency)	Cost of each unit.
_price_override_price	Override Price	String	Override the original unit price.
_part_units	Part Units	String	Number of part units.
_model_name	Model Name	String	Name of the model.
_model_segment_name	Model Segment Name	String	Name of the segment the model belongs to.
_model_product_line_name	Model Product Line Name	String	Product line to which the model belongs.
_model_product_line_attributes	Model Product Line Attributes	String	Product line attributes of Product line to which the model belongs.
_model_base_price	Model Base Price	Numeric (currency)	Base price of the model.

Variable Name	Label	Data Type	Description
_model_attributes	Model Attributes	String	Place holder for all model attributes.
_model_supplier_company_name	Model Supplier Company Name	String	Name of the model supplier company.
_model_bom	Model BOM	String	Link to the model BOM.
_model_variable_name	Model Variable Name	String	The variable name of the model.
_model_product_line_var_name	Model Product Line Variable Name	String	The variable name of the product line to which the model belongs.
_config_attributes	Config Attributes	String	Configurable Attribute is a composite attribute. *This attribute is always read only. However, Reconfigure action may be used to modify the values.*
_config_attr_text	Config Attribute Text	String	Configurable Attribute Text format is similar Configurable Attribute. *The only difference is that the values correspond to the display text of the attributes.*
_part_number	Part Number	String	Returns a part number from the parts database.
_part_desc	Part Description	String	The description associated with the part number in the parts database.
_part_base_price	Part Base Price	Numeric (currency)	The price of the part from the parts database.
_part_lead_time	Part Lead Time	String	Lead time associated with the part from the parts database.
_part_supplier_company_name	Part Supplier Company Name	String	Name of the part supplier company.
_serial_number	Serial Number	String	Serial Number
_serial_number_desc	Serial Number Description	String	Serial Number Description
_serial_number_custom_fields	Serial Number Custom Fields	String	Serial Number Custom Fields
_serial_number_custom_field_text	Serial Number Custom Field Text	String	Serial Number Custom Field Text
_price_subtotal	Subtotal	Numeric (currency)	Subtotal of line items. **See note below.**
_price_discount_percent	Discount %	Float	Discount in percentage.
_price_total_discount	Total Discount	Numeric (currency)	Total discount amount.
_config_attr_info	Config Attribute Information	String	Populated with variable name, attribute name, data type and menu type information.
_price_net_price	Net Price	Numeric (currency)	Net price = subtotal - total discount. **See note below.**
_price_override_discount	Override Discount	String	Override the original discount.
_is_line_item_mandatory	Is Line Item Mandatory?	Boolean	Used to identify if the line item is a mandatory spare.
_price_configurable_price	Configurable Price	Numeric (cur-	Original price as per catalog *(inclusive of discount and premium)*.

Variable Name	Label	Data Type	Description
		rency)	
_price_component_prices	Component Prices	String	Additional prices based on configuration.
_original_document_number	Original Document Number	String	Used to identify the document number from which the line item is copied.
_parts_extended_desc_1	Parts Extended Description 1	String	Extended description of the part.
_line_item_comment	Line Item Comment	String	Stores the comment for the line item.
_group_sequence_number	Group Sequence Number	String	Group sequence number for the sub-document.



The system line-level attribute, **Subtotal**, can now be derived from 2 separate places:

1. Any pre-existing commerce process that has List Price set as the default will mimic legacy functionality.
2. Unit Price: Any time a new Price Attribute Set is created, Unit Price will be the default.

Since *Net Price* = *Subtotal - Discount*, **Net Price** will also be affected by where you derive the subtotal.

Composite Attributes

Composite attributes are used to display one or more fields belonging to a family of attributes. This attribute type is unique in that you can control the display of composite attributes using XSL Views. Composite attributes can be used to display configurable attributes, product line attributes, model attributes and component pricing.

Composite Attribute	Description
Model Attributes	This type of composite attribute can be viewed within the Model Attribute Set on commerce sub-documents. It includes the regular attributes defined by the fullaccessuser in Product Definition to categorize basic characteristics of models within a single product family (segment).
Model Product Line Attributes	This is a composite attribute container for regular product line attributes and can be viewed within the Model Attribute Seton commerce sub-documents. The values for these attributes are those defined for the product line to which a model belongs. This composite attribute is exactly the same as the Product Line Attributes composite attribute, however it is administrated from within the Model Attribute Set on commerce sub-documents.
Product Line Attributes	This is a composite attribute container for regular product line attributes and can be viewed within the Product Line Attribute Set on commerce sub-documents. (These attributes are the same as those categorized by the Model Product Line Attributes composite attribute.) Regular product line attributes are defined by fullaccessusers in Product Definition to categorize basic characteristics of product lines within a single product family (segment).
Segment (Product Family) Attributes	This is a composite container that may be used in the future. Regular segment-level attributes are not supported on the system, so this composite attribute must NOT be used.

Composite Attribute	Description
Component Prices	Component Prices is a composite attribute that controls whether configured pricing details appear in commerce documents and in the document XML. This type of composite attribute is part of the Price Attribute Set.
Configurable Attributes	The Configurable Attributes composite attribute includes the configurable attributes associated with a specific model line item page on a transaction. Configurable attribute values are defined by users during product configuration. Configurable Attribute Information is populated with variable name, attribute name, data type and menu type information.

Attribute Format

Example: `Var_name~Name~Value~Display_text~Data_type~Menu~Multiple~Locked~Hide|&ldots;|&ldots;`

Format Part	Description
`var_name`	**Variable Name:** indicates variable name of the configurable attribute.
`name`	**Name:** indicates the name of the configurable attribute.
`value`	**Value:** indicates the value of the menu label or user-entered data.
`display_text`	**Display Text:** indicates the menu item display text. For text fields this will be blank.
`data_type`	**Data Type:** indicates if the configurable attribute is a text, float or integer.
`isMenu?`	**Menu:** indicates if it is a text field or menu type.
`isMultiple?`	**Multiple:** indicates if it is a multi-select configurable attribute.
`isLocked?`	**Locked:** indicates if it is a set type of configurable attribute.
`isHidden?`	**Hide:** indicates if this attribute is hidden.
`isArrayAttr?`	**Array Attribute:** indicates that attribute is an array type.

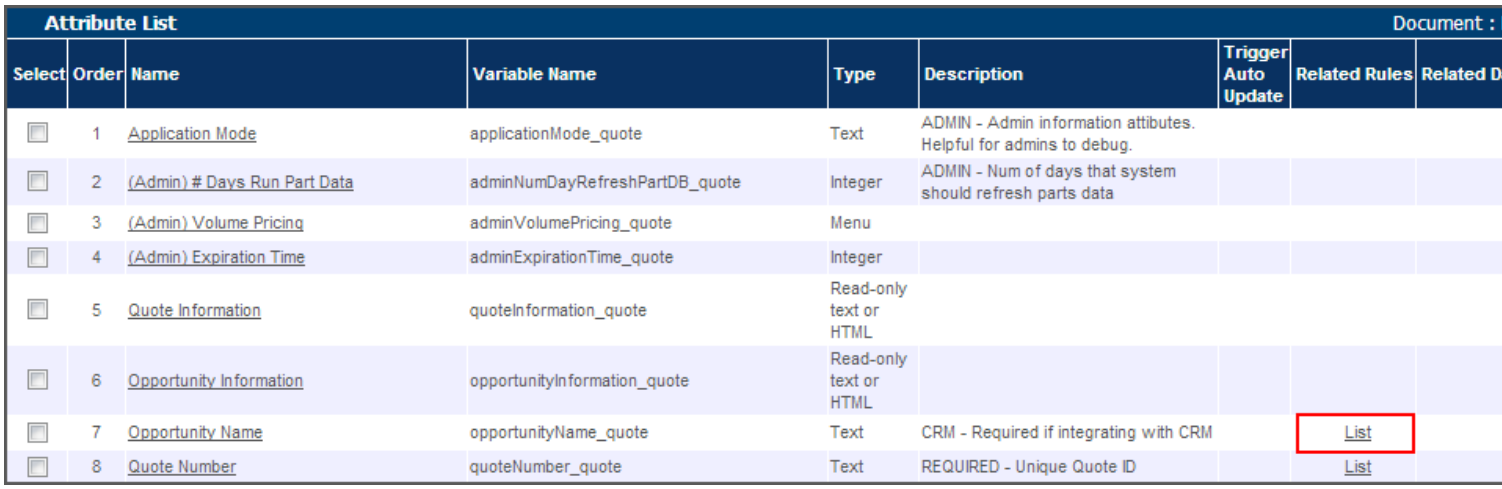


Do not use the reconfigure action to modify the values of the above attributes although it is possible to do so. *This is NOT recommended as this attribute is populated with system generated values. It could lead to unpredictable behavior in configuration and commerce.*

Viewing Related Rules

You will see all your related rules on this page and can do the following here.

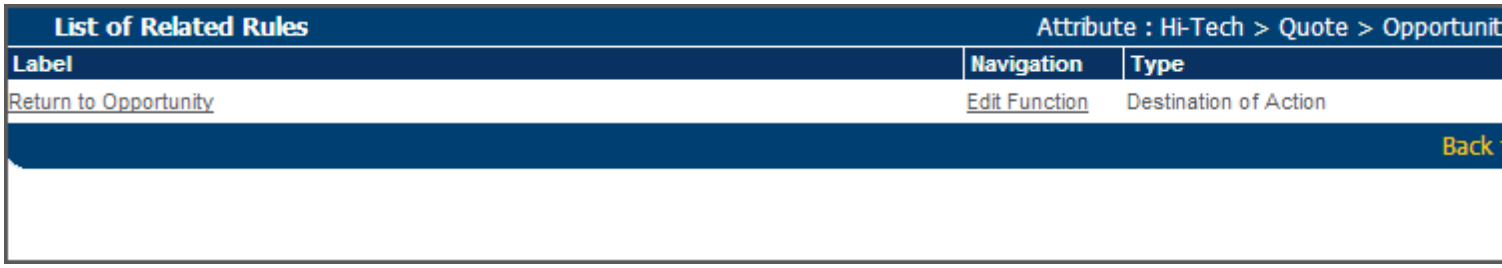
1. Navigate from: Documents > List (select Document) > Attributes > List



Attribute List — Document :

Select	Order	Name	Variable Name	Type	Description	Trigger Auto Update	Related Rules	Related D
☐	1	Application Mode	applicationMode_quote	Text	ADMIN - Admin information attibutes. Helpful for admins to debug.			
☐	2	(Admin) # Days Run Part Data	adminNumDayRefreshPartDB_quote	Integer	ADMIN - Num of days that system should refresh parts data			
☐	3	(Admin) Volume Pricing	adminVolumePricing_quote	Menu				
☐	4	(Admin) Expiration Time	adminExpirationTime_quote	Integer				
☐	5	Quote Information	quoteInformation_quote	Read-only text or HTML				
☐	6	Opportunity Information	opportunityInformation_quote	Read-only text or HTML				
☐	7	Opportunity Name	opportunityName_quote	Text	CRM - Required if integrating with CRM		List	
☐	8	Quote Number	quoteNumber_quote	Text	REQUIRED - Unique Quote ID		List	

2. Click **Edit Function** to open a function editor with the relevant script. **OR**



List of Related Rules — Attribute : Hi-Tech > Quote > Opportunit

Label	Navigation	Type
Return to Opportunity	Edit Function	Destination of Action

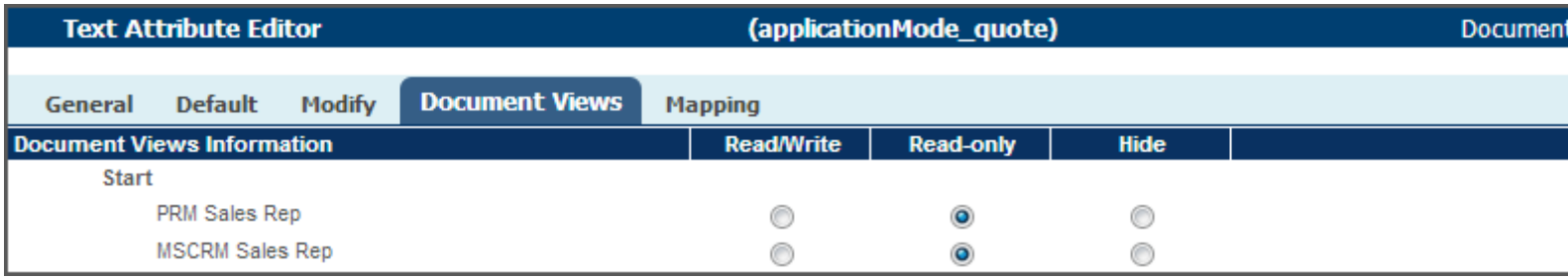
Back

3. Click the **Label** link (in this example, it would be "Return to Opportunity") to open the relevant document action page and click **Define Destination Rule** to open a function editor with the relevant script.

Document Views Tab

Navigation: Admin Home Page > Commerce Processes > Process Definition > Documents > Attributes

The Document Views Tab allows the fullaccessuser to control the access level of an attribute to users in different steps/ profiles of the transaction.The fullaccessuser can change the attribute access levels to the following in a certain step/ participant profile:



Text Attribute Editor (applicationMode_quote) — Document

General | Default | Modify | **Document Views** | Mapping

Document Views Information	Read/Write	Read-only	Hide
Start			
PRM Sales Rep	○	◉	○
MSCRM Sales Rep	○	◉	○

Access Level	Description
Read-Only	The attribute cannot be edited in this step or participant profile for the current user.

Access Level	Description
Read/Write	The attribute value can be edited in this step or participant profile for the current user.
Hide	The attribute will not be displayed in this step or participant profile to the current user.

Main Document Attribute Sets

Attribute sets contain more than one commerce attribute. There are a number of attribute sets you can include in your documents, including credit card attribute sets, address attribute sets, product line attribute sets, etc. Each attribute set is made up of a group of individual attributes. The individual attributes can be customized, or hidden from the set. All attribute sets are created together or removed together.

Price Book Attribute Set

A price book attribute set allows you to store price book information for all parts in a transaction. Only one price book attribute set can be added per sub-document. A transaction can support only one price book. When you create a price book attribute set, the commerce system automatically generates several fields, each of which has editable characteristics.



Price Book Attribute Sets are generated automatically.

Bill To/Ship To Attribute Set

Bill To/Ship To attribute sets can be created only once on every main document. When you create an address attribute set, the commerce system automatically generates 26 fields, each of which has editable characteristics.



Bill To/Ship To Attribute Sets are generated automatically.

Credit Card Attribute Set

Credit card attribute sets can be created only once on every main document. When you create a credit card address attribute set, the commerce system automatically generates three fields, each of which has editable characteristics.

Field	Description
Credit Card Type	Text field with the variable name `_cc_type`
Credit Card Number	Text field with the variable name `_cc_number`
Credit Card Expiration Date	Text field with the variable name `_cc_exp_date`

When the user is expected to populate the Credit Card Attribute fields, he will be presented with only one action. The user should not be allowed to progress to any other part of commerce without clicking this action. *The action will:*

- Send the Credit Card Attribute set to the third-party Credit Card payment processor. This will most likely occur in the Advanced Modification on the action, using the `urldatabypost` function.
- In the Advanced Modification script, the Credit Card Attribute fields will be obfuscated (i.e., changed from "1234567812345678" to "**********5678"). *This is done because once the script completes, the value that is assigned to this attribute will get saved in the transaction XML. Ensuring that credit card fields are concealed assures that BigMachines does not store credit card data.*



Credit Card Attribute Sets are generated automatically.



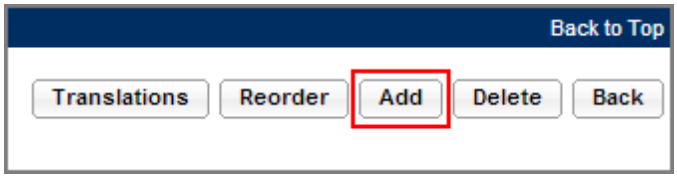
The credit card expiration date is tested to make sure that it is a future date. The credit card number is tested against the type entered to ensure that the numbers meet basic validity for that credit card type.

Example: Visa credit card numbers conform to a specific testable algorithm without calling an outside credit card processor. This does not validate that it is a current, active credit card; just that it conforms to the numeric algorithm for that credit card type. *It could still be denied by the credit card processor, but BMI ensures that the user didn't input the number incorrectly.*

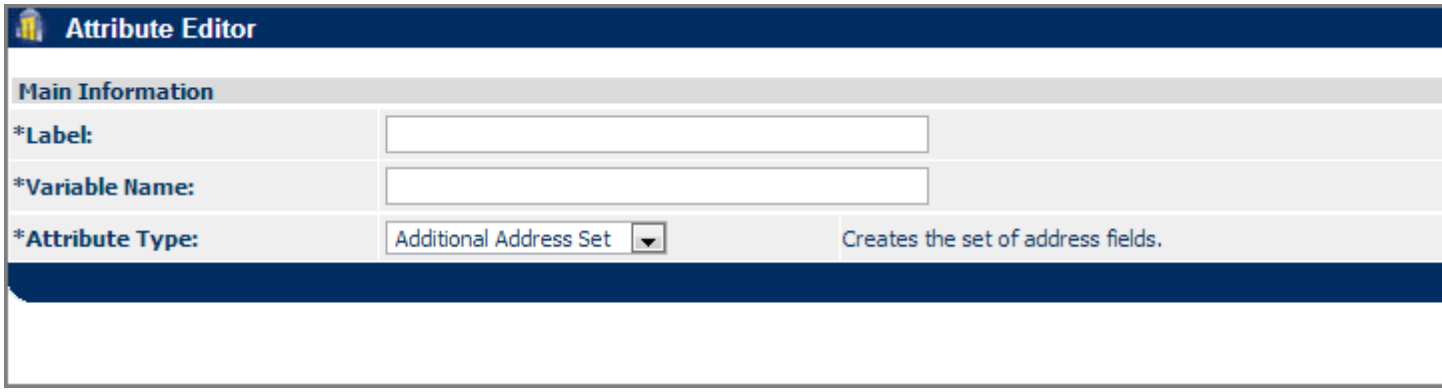
Additional Address Attribute Sets

Additional Address Sets can be created multiple times on every main document. When you create additional address attribute sets, the commerce system automatically generates 13 fields, each of which has editable characteristics. **Creating an Additional Address Attribute Set:**

1. Navigate from: Process Definition > Documents > Quote > Attributes > Attribute List
2. Click **Add** *(located at the bottom of the Attribute List page)*.



3. Enter a **Label** and **Variable Name** for the set.



4. Select **Additional Address Set** as an **Attribute Type**.
5. Click **Add**.

Click the **Additional Address Attribute Set** to edit individual attributes contained in a set.

Sub-Document Attribute Sets

Attribute sets contain more than one commerce attribute. There are a number of attribute sets you can include in your documents, including credit card attribute sets, address attribute sets, product line attribute sets, etc. Each attribute set is made up of a group of individual attributes. The individual attributes can be customized, or hidden from the set. All attribute sets are created together or removed together.

Part Attribute Set

Part Attribute Sets can be created only once on every sub-document. When you create an address attribute set, the commerce system automatically generates about 41 fields, each of which has editable characteristics (except for variable names).

Part Attribute Sets are generated automatically.

There are two special types called 'Link to Dialog Window' for the Part Extended Description Fields. These types create a link in the transaction to a dialog window to display content. You can edit the link text on the editor page for this type.

Serial Number Attribute Set

A serial number attribute set allows you to store serial number information for each part line item in a transaction. Only one serial number attribute set can be added per sub-document. When you create a serial number attribute set, the commerce system automatically generates several fields, each of which has editable characteristics.



Serial Number Attribute Sets are generated automatically.

Model Attribute Set

A model attribute set contains all the details of the current catalog line item. Only one model attribute set can be added per sub-document. When you create a model attribute set, the commerce system automatically generates several fields, each of which has editable characteristics.



Model Attribute Sets are generated automatically.



There is a special type called 'Link to Dialog Window' for the Model Bill Of Material field. This creates a link in the transaction to a dialog window to display the Bill of Material content for the model coming from configuration. You can edit the link text on the editor page for this type.

Config Attribute Set

A config attribute set contains all the configurable attribute details of the current catalog line item. Only one config attribute set can be added per sub-document. When you create a config attribute set, the commerce system automatically generates three composite fields, each of which has editable characteristics.



Config Attribute Sets are generated automatically.

Price Attribute Set

A price attribute set allows you to store all pricing related data for a line item. Only one price attribute set can be added per sub-document. When you create a price attribute set, the commerce system automatically generates several fields, each of which has editable characteristics.



Price Attribute Sets are generated automatically.



Unit Price has been changed to **List Price**. List Price still behaves like it did in previous versions. The variable name remains: _price_list_price_each.

Unit Price is now the FINAL price, prior to quantity and discounting. This will take all previous pricing (including customer-specific pricing) into account.

Credit Card Attribute Set

Credit card attribute sets can be created only once on every main document. When you create a credit card address attribute set, the commerce system automatically generates three fields, each of which has editable characteristics.

Field	Description
Credit Card Type	Text field with the variable name `_cc_type`.
Credit Card Number	Text field wiith the variable name `_cc_number`.
Credit Card Expiration Date	Text field field with the variable name `_cc_exp_date`.

When the user is expected to populate the Credit Card Attribute fields, he will be presented with only one action. The user should not be allowed to progress to any other part of commerce without clicking this action. *The action will:*

- Send the Credit Card Attribute set to the third-party Credit Card payment processor. This will most likely occur in the Advanced Modification on the action, using the `urldatabypost` function.
- In the Advanced Modification script, the Credit Card Attribute fields will be obfuscated (i.e., changed from "1234567812345678" to "**********5678"). *This is done because once the script completes, the value that is assigned to this attribute will get saved in the transaction XML. Ensuring that credit card fields are concealed assures that BigMachines does not store credit card data.*



Credit Card Attribute Sets are generated automatically.

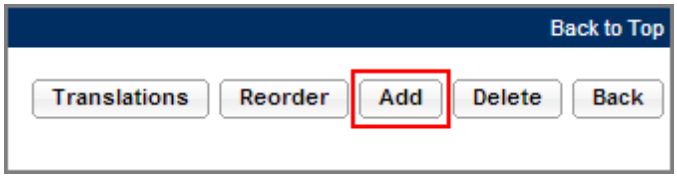
The credit card expiration date is tested to make sure that it is a future date. The credit card number is tested against the type entered to ensure that the numbers meet basic validity for that credit card type.

Example: Visa credit card numbers conform to a specific testable algorithm without calling an outside credit card processor. This does not validate that it is a current, active credit card; just that it conforms to the numeric algorithm for that credit card type. *It could still be denied by the credit card processor, but BMI ensures that the user didn't input the number incorrectly.*

Product Line Attribute Set

A product line attribute set contains all the details of the product line to which the catalog line item belongs. Only one product line attribute set can be added per sub-document. When you create a product line attribute set, the commerce system automatically generates five fields, each of which has editable characteristics.

1. Navigate from: Process Definition > Documents > Line Item > Attributes > Attribute List
2. Click **Add** *(located at the bottom of the Attribute List page)*.



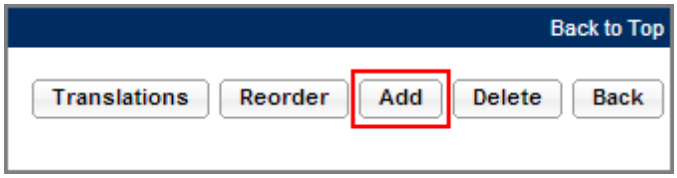
3. Select **Product Line Attribute Set** from the Attribute Type drop-down menu.



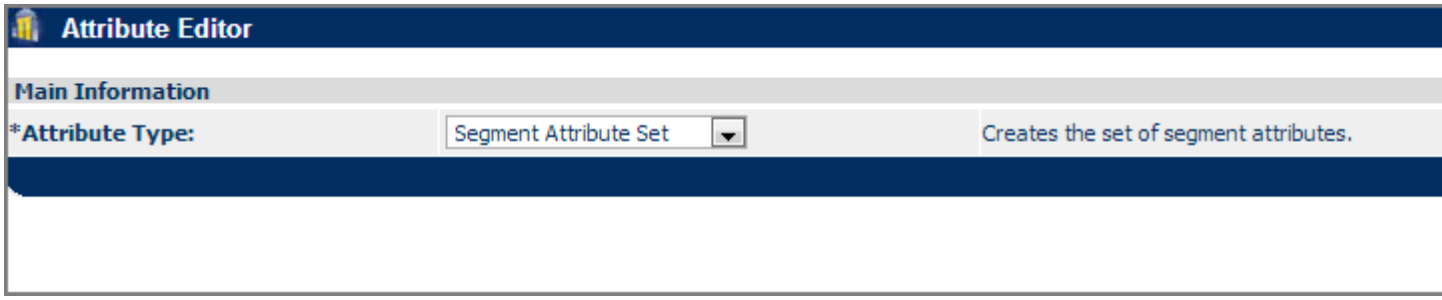
4. Click **Add**.

Segment Attribute Set

1. Navigate from: Process Definition > Documents > Line Item > Attributes > Attribute List
2. Click **Add** *(located at the bottom of the Attribute List page)*.



3. Select **Segment Attribute Set** from the Attribute Type drop-down menu.

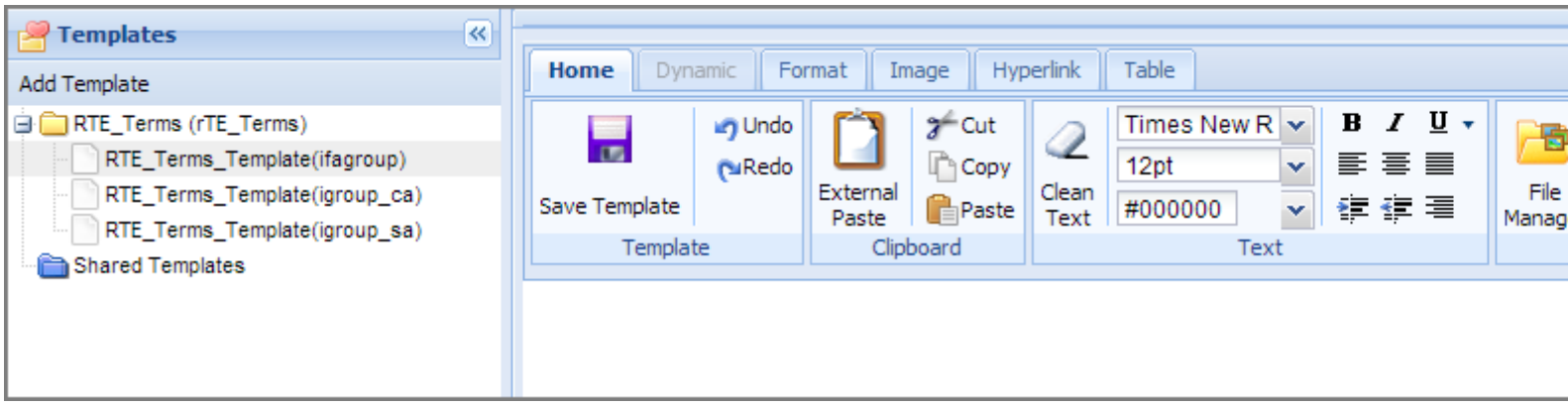


4. Click **Add**.

Model BOM, **Part Extended Description 1**, and **Part Extended Description 2** are of a special type called "Links to a Dialog Window". This links the transaction to the information coming from configuration.

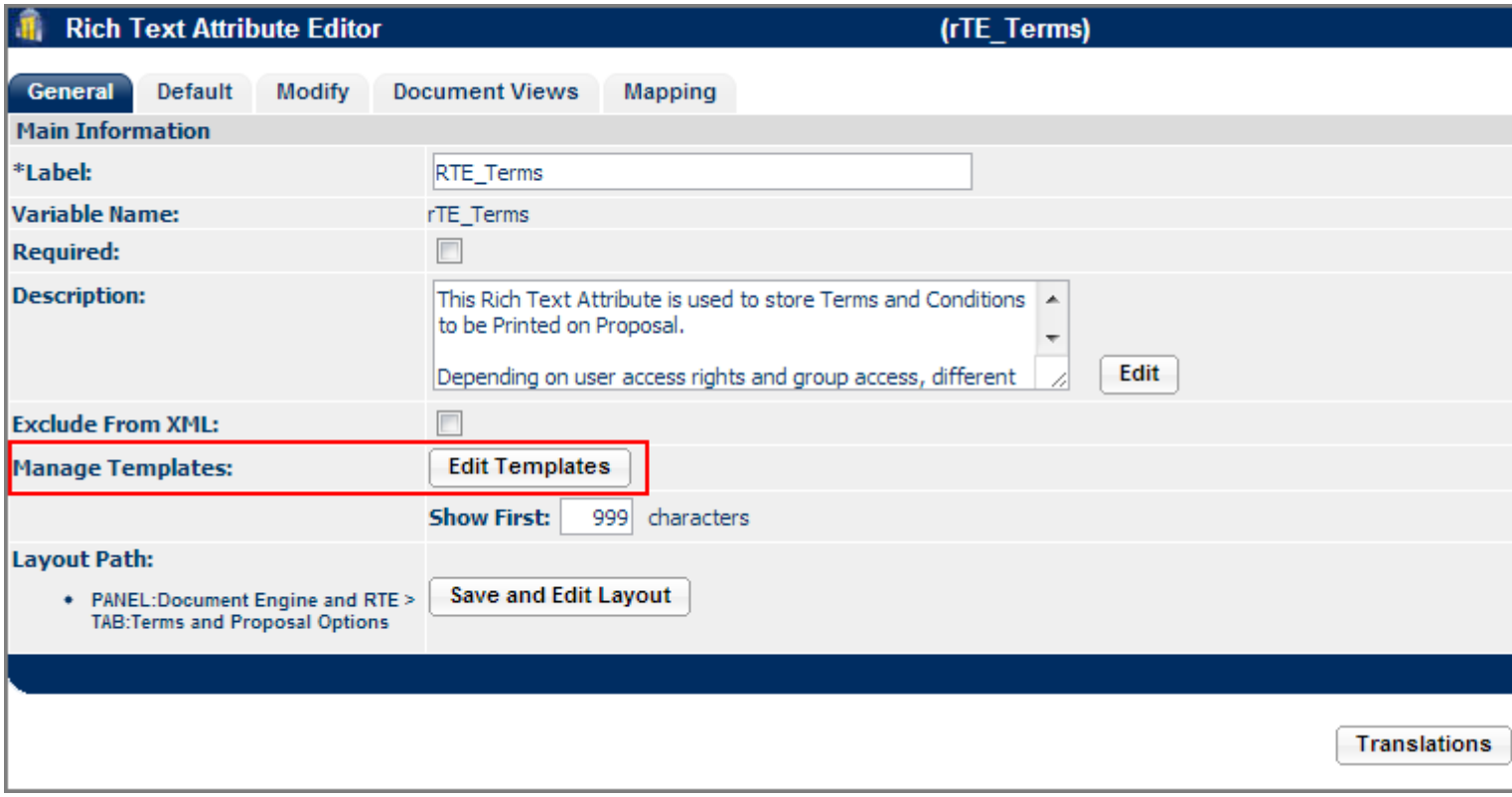
Rich Text Editor

The Rich Text Editor allows the user to edit rich text attributes and associated templates.

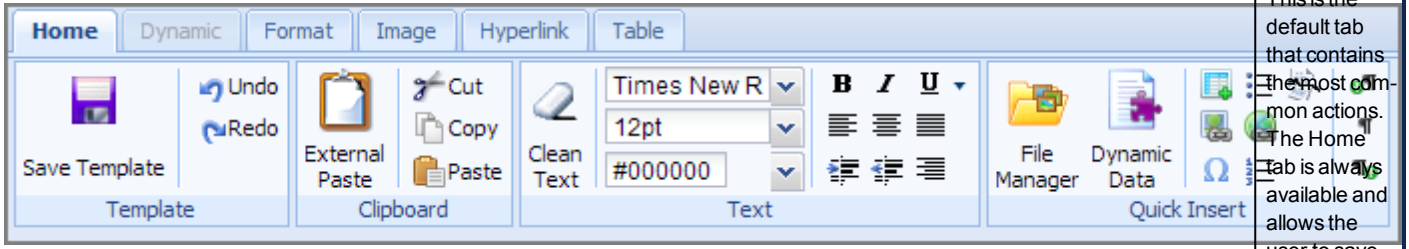


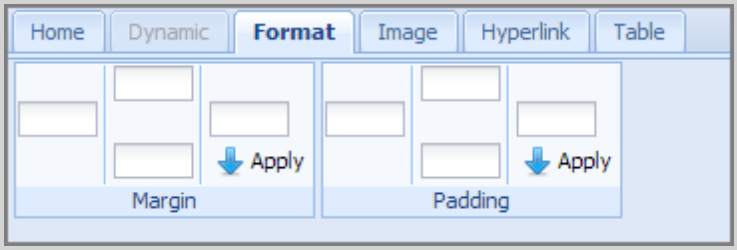
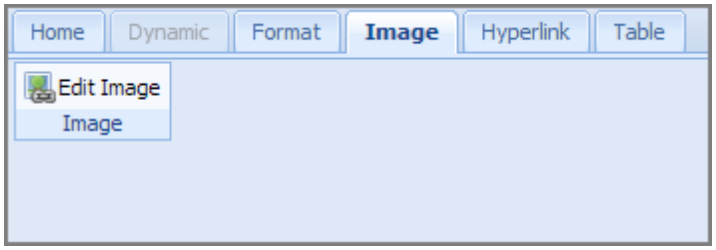
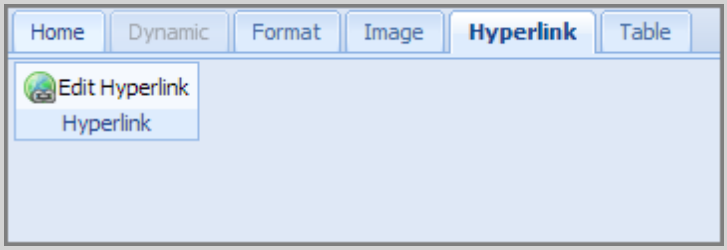
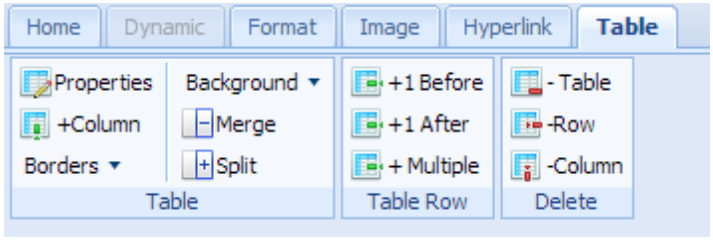
Navigation

Navigate from: Admin > Process Definition > Documents > Attributes > Attribute List > Rich Text Attribute > **Edit Templates**



Sections	Description
Content Editor Window	The user can edit content in this section. The actions in the Ribbon act on the selected content within this window.
Template Tree	The Template tree gives the user access to the already created templates. To add a template, the user has to navigate to the correct template in the Template tree and click **Insert Into Text**. *The user can then save the page as a new template.*
Template Preview	The Template Preview section allows the user to preview existing templates. The user navigates to the desired template on the Template tree and then selects the template. *This lets the user preview the template in the Template Preview section.*
Ribbon	The actions on the Ribbon are grouped in tabs. Clicking a tab switches the tool bar actions to a different set of actions.

Tab	Example	Description
Home		This is the default tab that contains the most common actions. The Home tab is always available and allows the user to save

Tab	Example	Descrip-tion
		the template.
Format		This allows users to format appearance of margins and padding.
Image		The actions in this tab can be used to change the source URL or the height and width of the image.
Hyper-link		The user can use this tab to edit the hyperlink.
Table		The Table tab allows the user to cus-tomize tables in a page.

Text Library

Text Library stores the predefined templates that are used in Rich Text Attributes. *Templates can be stored in two kinds of folders:*

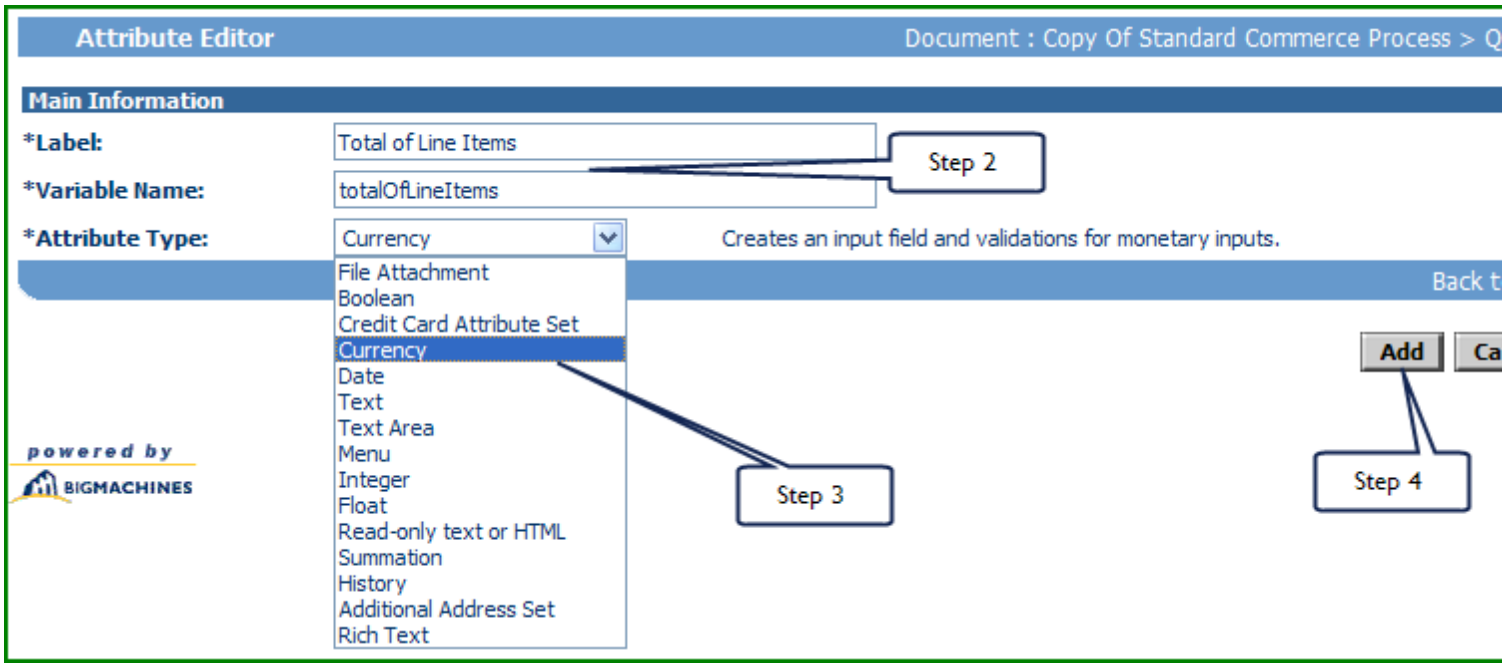
- **Shared Templates:** The templates that reside in this folder can be shared across Rich Text Attributes within a process. *Only the superuser can save templates in this folder.*
- **Individual Folder for each Rich Text Attribute:** The templates that reside in the individual folders for each Rich Text Attribute can only be used for that particular attribute by the template . *The template can only be edited by the user that created it. A different user can however, save the template and then edit the saved template.*

ADMINISTRATION

Adding Attributes to a Document

By default, the document attributes are listed on the Attribute Lists page in the order that they are created (the composite attributes are always listed last).

1. Click **Add** on the Attributes List page
2. Enter the **Label** and **Variable Name** of the attribute.



3. Choose the **Attribute Type** from the drop-down menu.
4. Click **Cancel** to return to the Attributes List page without saving changes or click **Add** to save changes and open the Attribute Editor. *The other fields on the Attribute Editor page depends on the attribute type.*

After creating document attributes, you can select one of the attributes to set a tab name for the commerce process. *Tabs are used to designate separate transactions on the Commerce Manager list page.*

Variable Names cannot be edited after they are created.

Editing a Document Attribute

1. Choose the attribute or attribute set you want to edit on the **Attributes List** page. *Do one of the following:*
    - **To edit an individual attribute:** Click the attribute name.
    - **To edit an attribute set:** Click the attribute set name. Then click the attribute you want to edit.
2. Edit any of the fields displayed in the **Attribute Editor**. *To edit the default value, access rights, layout or product mapping, click a tab name and edit the information displayed.*
3. Click **Apply** to remain on the current page or **Click Update** to return to the **Attributes List Page**. Click **Cancel** to return to the Attributes List page without saving changes.



Document attributes can be edited at any time.



Edits are not visible until the process is deployed (or re-deployed).



Previously created variable names appear read-only because they are not editable.

Deleting a Document Attribute

1. Select the checkbox next to the desired attribute.
2. Click **Delete**.
3. If you want to delete an entire attribute set, click into the `<attribute_set_name>` and click **Delete All**.



If you delete attributes post-deploy, please make sure to re-map and repopulate your data columns for the transaction manager to work correctly.



The remaining attribute(s), if any, will be automatically reordered, when the page reloads.



It is not possible to recover the deleted data. *When an attribute is deleted, it is simultaneously deleted from all document views.*

System Attributes cannot be deleted; nor can you delete individual attributes from within an attribute set.

Ordering Document Attributes

Order document attributes to set how they will appear in the document.

1. Click **Reorder** on the Attributes List page.
2. Change the order number on the attribute you want to reorder.
3. Click **Apply** to remain on the current page/click **Update** to return to the Attributes List Page. *The new attribute order displays on the Attributes for Document page.*



You can reorder individual attributes, including the individual attributes contained within an attribute set.



Order changes take effect after the commerce process is deployed (or re-deployed).

Order of Operations: Rule Precedence

1. Config Flow Rules
2. Constraint Rules
3. Recommendation Rules
4. Constraint Rules (again)
5. Hiding Rules
6. Recommended Item Rules
7. BOM Rules

Attribute Order & Rule Precedence

1. The attribute order affects rule order in the following ways:
2. Config rules with "True" conditions are reviewed.
3. The rules that reference the first Configurable Attribute are run.
4. The rules that reference the second Configurable Attributes are run. *Rules on subsequent attributes are run.*



Order becomes important when there are dependent Recommendation Rules that need to be calculated in a specific sequence. Example: D is the sum of B and C; C = A*2. If C appears after D within the Attribute List, D's calculation will use a stale value for C, resulting in an incorrect value for D.

Setting a Default Value for an Attribute

1. Navigate from: Commerce Processes > Process Definition > Select a Process > Documents > Attributes > Default Tab
2. Select the **Default** tab.
3. Click one of the following radio button options:

Radio Button Options	Description
None	To not specify a default value, enable the None radio button option.
System Variable	To set a default value based on a system attribute, enable this radio button option and choose the system attribute you want to auto-populate the field.
Data	To set a static value, enable this radio button option and enter a default value.
Define Function	To define a dynamic default value, enable this radio button option and click Define Function. Use this advanced function editor to create a function that sets the default value into the field.
Select a Template	*This is only an option for Rich Text type attributes.* If this option is chosen, then the rich text attribute is pre-populated with the selected template.

4. Click **Apply** to remain on the current page or **Update** to save changes and return to the Attributes List Page.



Defaults cannot be set for the following types of actions: Summation, Read-Only Text, HTML File Attachment, Composite.

Setting an Access Value for an Attribute

1. Click on the **Document Views** Tab.
2. Select one of following **Access Level** radio button options:

Access Level	Description
Read-Only	The attribute cannot be edited in this step or participant profile for the current user.
Read/Write	The attribute value can be edited in this step or participant profile for the current user.
Hide	The attribute will not be displayed in this step or participant profile to the current user.

3. Click **Apply** to remain on the current page or **Update** to return to the Attributes List Page.

Hiding Attributes

1. Click the **Mapping** tab on the Attribute Editor.
2. Do one of the following:
   - *To hide the document attribute (or attribute set) in documents created for parts, models, product lines or segments:* select one or more checkboxes.
   - *To hide the commerce attribute (or attribute set) in documents created for specific segments:* select the **Hide For Segments** checkbox, and highlight the segments wherein purchases should not include the attribute.
3. Click **Apply** to save changes and remain on the page or **Update** to save changes and return to the Attributes List page.



This functionality enables you to hide attributes in documents created for a single type of item.

Example: if an attribute is hidden for parts, then the attribute would not appear in documents whose line items are for parts. However, if the line items are for parts and models, attributes marked ”hide for parts” would appear.

Adding Commerce System Variables to Documents

You can display commerce system attributes on documents or make them hidden and only use them in document calculations. When commerce system attributes are mapped to document attributes, the variable values are determined once at document creation time. To gain access to system attribute values at other times in a document's existence, you can return them in functions associated with modify type document actions.

1. Create a document attribute. *Most commerce system variables are string types that can be quickly displayed using a text attribute.*

2. To associate a system variable value with the attribute, do one of the following:
    - **For text attributes:** After defining the attribute name, variable name and attribute description, click the **Default** tab and enable the **System Variable** radio button. Then highlight the system variable you want to display.
    - **For numeric attributes:** After defining the attribute name, variable name, attribute description and any validation fields, click the **Default** tab and enable the **Function** radio button. Click the **Define Function** button, mark the system variable you want to display, scroll to the bottom of the window and click **Next**. In the **Editor** window, enter: `return (<_variable_name>);`
3. Click **Finish**.

XML Example

```
<current_date><![CDATA[ 05/31/2002 12:54PM ]]></current_date>
<supplier_base_currency><![CDATA[ USD ]]></supplier_base_currency>
<user_currency_pref><![CDATA[ AUD ]]></user_currency_pref>
<currency_conv_rate><![CDATA[ 1.79624 ]]></currency_conv_rate>
```



You can use numeric attribute types to store commerce system variables as long as the variable value is converted from string type to numeric type.



When commerce system variables are included in a document, the system variable values are also included in the XML generated for the document. The values appear in XML tags named with the attribute variable names used to store system variable values.

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*

        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*

    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



Summation, composite and read-only text or HTML attributes are not editable.

Attribute range calculations are validated on the Value Punch-In. **Example:** if the value was passed in the URL to the Configuration, it will be validated and an error message will appear. *This affects both Mobile and Desktop.*

Zero and blank integer values are treated as separate values:
-Zero = 0
-Blank = ""



You cannot change the type of a menu attribute if it is already on the layout.

User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

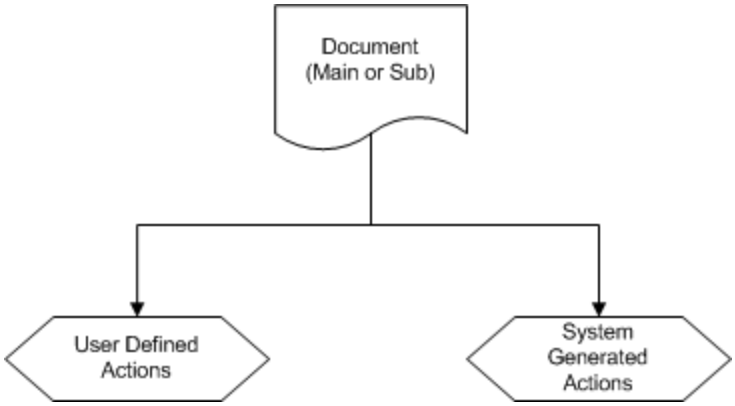
Never create a Summation Attribute at the line level.

RELATED TOPICS

DOCUMENT ACTIONS

OVERVIEW

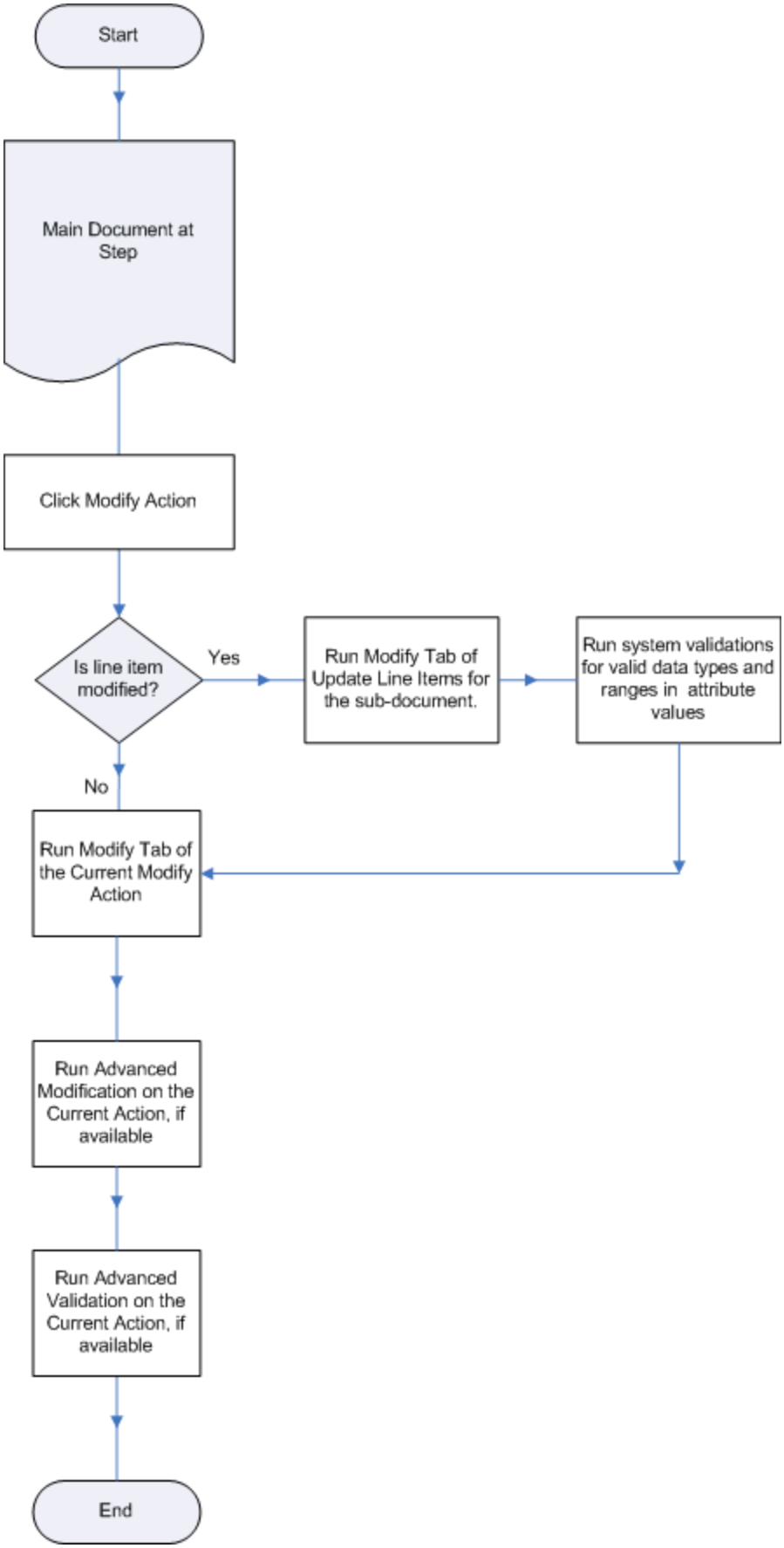
Document actions trigger events within documents to occur. The commerce system creates a default set of actions for use in each document, and generates additional actions when commerce documents include certain attribute types. Both main documents and sub-documents can consist of system generated actions as well as user-defined actions, created based on supported action types. Because Actions are required to carry out operations based on the business logic, they are comprised of components.



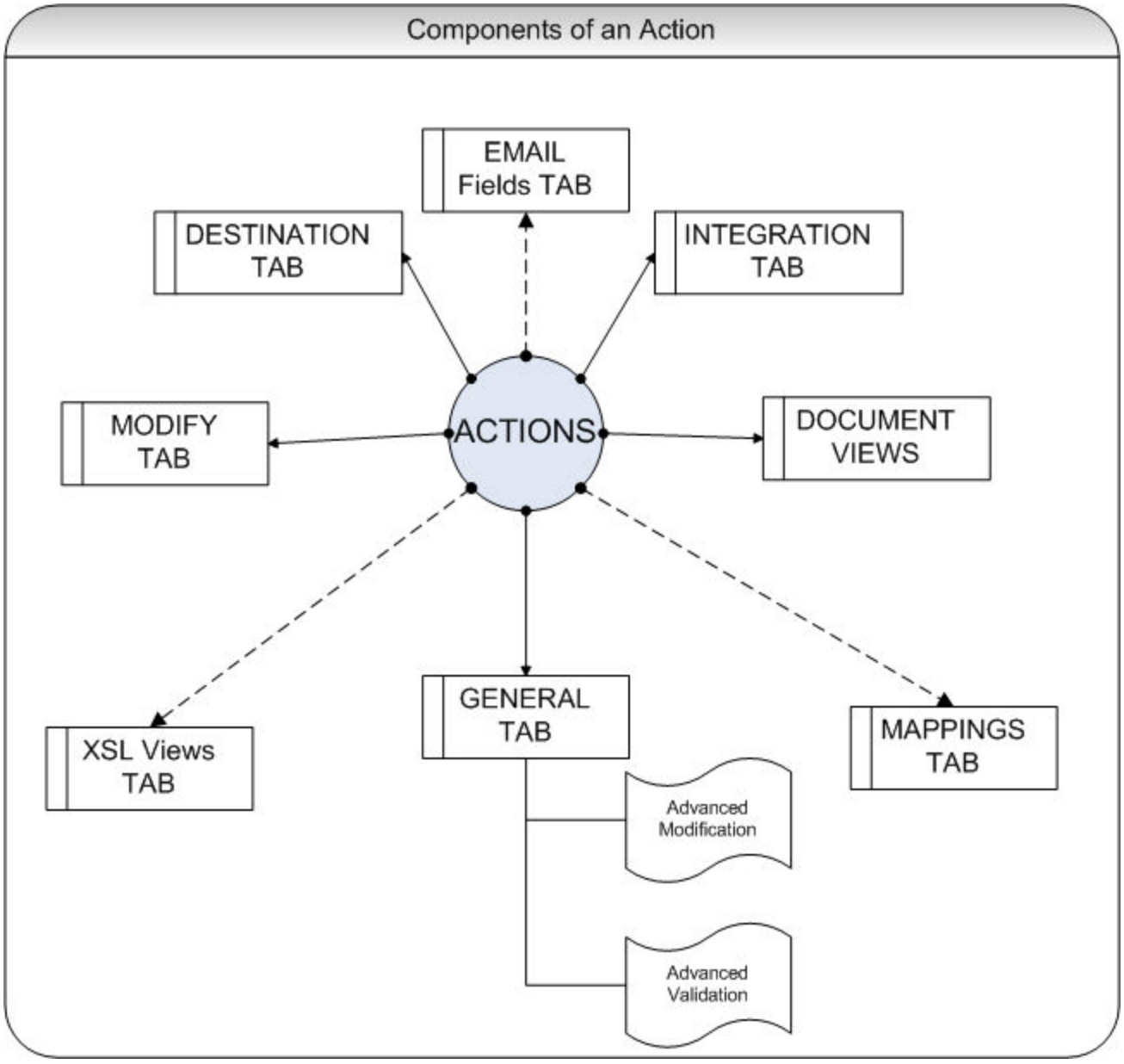
All document actions are administrated from the Actions for Document page. You can customize system-generated actions by modifying action labels and creating action rules. In addition to the system-generated action defaults, you can also create custom actions for the purpose of saving document data, directing the system to a different page, offering print or email options, and showing the change history for a document. Users usually perform document actions, but timers can be set up so the system can automatically perform document actions after specified periods of time.

Action Display Types	Description
Buttons	Buttons signify most actions like send, respond, update, etc.
Hyperlinks	Hyperlinks designate specific actions, like `r;open sub-document` and `r;view Account data`.

Order of Operations


Start
Main Document at Step
Click Modify Action
Is line item modified?
Yes
Run Modify Tab of Update Line Items for the sub-document.
Run system validations for valid data types and ranges in attribute values
No
Run Modify Tab of the Current Modify Action
Run Advanced Modification on the Current Action, if available
Run Advanced Validation on the Current Action, if available
End

Components (Tabs)



User-Defined Document Actions

fullaccessusers can define actions in a commerce process. All the action types below are common for the main document and sub-document.

Action Name	Description
Modify	Most common type of user defined action. A modify type action can be entirely customized to suit the fullaccessuser's needs.
Back	Used to take the user back, depending on which document they are on. If the user is on the sub-document, clicking Back will take the user to the main document. This action does not save user data.
Add from Catalog	Allows the user to add line items to a transaction from the main document. Add from catalog actions take a user to the home page from where the user can navigate to specific catalog items/parts/favorites and add desired items back to the transaction. Destination rules can be set for Add from Catalog type actions. Multiple Add from Catalog type actions can be added to the same document to offer users different catalog destinations within the application. *This action has different options for the main document and sub-document:* **Main Document:** Here the tab contains a simple URL field. The valid value for the URL field is the relative path to any page on the BigMachines application. Example: on the main document, you can use an Add from Catalog action as a quick link that always goes back to the same page, either a page in the catalog or a Favorites List page. **Sub-Document:** Here the tab contains a simple URL field as well as an advanced function option. On sub-documents, you can set up advanced destination rules for Add to Catalog type actions. This has the added benefit of enabling you to link the Add from Catalog action to different catalog pages depending on the subsystem components that still need to be added to the document. The valid value for the URL field is the relative path to any page on your application. The advanced function is usually used to display the action in the line item section of the main document. Add from Catalog type actions created for sub-documents should be added to the line item group for the main document only. They should not be added to the sub-document view. *These actions are supported for the Favorites List page, the Commerce home page, and Catalog pages. Although this section can be directed to external URL's, it is not possible to return back to a commerce document from an external URL.*
Print	Enables users to access printer-friendly forms for printing. It is a modify type action and hence has all characteristics of a modify action, including document view, integration and the modify tab. *When associating a print action with an XSL view that outputs data as PDF, end users may have to configure a setting Adobe Acrobat to be able to view the resulting document. The setting turns off browser display for PDF documents.* **XSL Views:** The Print action contains a special tab called the XSL View tab. This action type must be used in conjunction with an XSL format file that contains the printer-friendly format you want to associate with that particular print action. The XSLs displayed on this tab come from the Printer Friendly/History XSL Views section. This action can be associated with one or more XSL views. If multiple XSL views are available, users can select a print form from the ones available.
E-mail	Enables users to access email-friendly document formatting. It is a modify type action and hence has all characteristics of a modify action, including document views, integration and the modify tab. *When associating an email action with an XSL view that outputs data as PDF, end users may have to configure a setting in Adobe Acrobat to be able to view the resulting document. The setting turns off browser display for PDF documents.* **XSL Views:** The email action contains a special tab called the XSL Views tab. This action type must be used in conjunction with an XSL format file that contains the email-friendly format you want to associate with that particular email action. The XSLs displayed on this tab come from the Printer Friendly/History XSL Views section. This action may be associated with one or more XSL views, from which you can choose one. **Email Fields:** The email action also contains a special tab called the Email Fields tab. Email field rules instruct the system to auto-populate fields in the email pop-up window. With the fields populated by the system, users can sent out emails quickly. Email field rules apply to the email document action only.
Display History	Used to extract the change history for a document. This action requires an XSL format file that controls how and which changes appear. **XSL Views:** The Display History action contains a special tab called XSL Views tab. The XSLs is displayed on this tab come from the Printer Friendly/History XSL Views section. This action can be associated with one or more XSL views. If multiple XSL views are made available with the action, users can select the email form they want to use.
Export Attachment	The Export Attachment action enables a user to add the proposal (PFF) to the related partner opportunity (only on an integrated application). It is a modify type action and hence has all characteristics of a modify type action including document views, integration, and the modify tab. **XSL Views:** This action contains a special tab called the XSL Views tab. This action type must be used in conjunction with an XSL format file that contains the printer-friendly format you want to associate with that particular action. The XSLs displayed on this tab come from the Printer Friendly/History XSL Views section. This action can be associated with one or more XSL views. If multiple XSL views are made available with the action, users can select the email form they want to use.
Copy Line Items	The Copy Line Items action can be defined only for the main document. This action enables users to make one or more copies of one or more line items (configured and non-configured) at a time. It is a modify type action. However, unlike other modify types actions, the Copy Line Item Action possesses an Initialization Tab and has no Integration tab.

Copy Line Items Order of Operations

1. Click **Copy** button.
2. The form **Data** is saved as is *(along with the line item group attribute values and other main document parameters on the form/document JSP page)*.
3. Run Initialization.
4. Run **Modify** tab of Copy Line Item action.
5. Run **Advanced Modification of Copy Line Item** action.
6. Run **Advanced Validation of Copy Line Item** action.

Splitting Line Items

1. Set quantity to one within the **Copy Line Item** action **Initialization** tab.
2. Select the option **Default to Quantity** within the **General Tab** to specify that the "Number of Copies" field in pop-ups is pre-populated with the desired quantity.



When the user selects the line item of quantity "n" and then clicks the **Copy Line Item** button, a pop-up with the number of copies with a pre-populated value "n" is displayed to the user.

When the user clicks the **Copy** button, "n" copied line items with quantity 1 (initialization tab will initialize the quantity to 1) is produced. If n=1, then the copy line items pop-up does not appear.

Implementing Copy Line Items in an Integrated Process

To successfully implement the **Copy Line Item** action in an integrated process, the user must set the opportunity's line item ID to **Revert to Default** in the initialization tab of the **Copy Line Item** action. *This is done to ensure that the copied line items are added to the SFDC opportunity with the unique line item ID's.*



You can set the maximum number of copies in the **General Tab** for this section.

When copying a configured line item, both mandatory and non-mandatory recommended items are copies, along with configured line item.

If the maximum number of copies is higher than 1, a pop-up appears upon clicking the **Copy Line Item** button. The user can specify the number of copes in this pop-up. *You can also specify that the Number of Copies field in the pop-up be populated with the quantity of the selected line item.*

You can map the line item attribute to the pop-up in the Admin Columns page.



Recommended items are copied, maintaining the ratio of the configured line item (model) to the recommended items. *You cannot copy a recommended item separately.*

Example: assume that a model `M1` is associated with a dozen of part `P1` (recommended item). If you make two copies of `M1`, then you will get two additional line items of `M1` and two additional dozens of part `P1`.



This action does not allow users to copy non-configured line items (part) if they do not have access to the required price books.

The **Copy To Favorites** action is available only on the main document. *It is a special action that is used to copy selected line items from the transaction to the favorites list. From here the user can add it back to the transaction.*

System-Generated Document Actions

Main-Document System Actions

Action Name	Variable Name	Description
Update Line Items	`_update_line_items`	Users can modify line item data on a main-document, then save the changes. This action is automatically created when a sub-document is linked to a main document.
Change Currency	`_change_currency`	Users can change the currency of the transaction to a selected currency from a drop-down menu. This action is automatically created when a main-document is created. This action will only appear to users on transaction that don't contain line items.
Remove [sub-document]	`_remove_<sub_document_variable_name>`	Users can remove the products that are displayed on the sub-document. This action is automatically created when a sub-document is linked to a main document.
Open [sub-document]	`_open_<sub_document_variable_name>`	Users can view product details on a line item within the sub-document. This action is automatically created when a sub-document is linked to a main document.
Create [next main document]	`_create_<next_main_document_variable_name>`	Users can move on to the next main document in the commerce process. This action automatically appears with all main documents.
Autofill	`_auto_fill_action`	Automatically links data from the Accounts database to the main document. *Users must enter an ID into the Customer ID field.* This action is automatically created (or removed) when a Customer ID attribute field is created (or removed).

Action Name	Variable Name	Description
Browse	`_browse_action`	Dialog box pops up that allows users to search the Accounts database for a customer record. When a user clicks 'Populate', after selected an accounts record, the corresponding accounts record data is auto-populated into the mapped account fields in the main document. This action is automatically created (or removed) when a Customer ID attribute field is created (or removed).
View	`_view_crm_action`	Allows suppliers to view an accounts profile when an ID appears in the Customer ID attribute field. This action is automatically created (or removed) when a Customer ID attribute field is created (or removed).
Select Alternate Address	`_select_alternate_address_action`	Host company users can associate an alternate Accounts address with a transaction. The system only displays this action when the Customer ID associated with a transaction has multiple addresses.
Refresh Quote	`refreshQuote`	Pulls data from the server, but does not save it. While doing so, it will not perform a modification.

Sub-Document System Actions

Action Name	Variable Name	Description
Reconfigure	`_reconfigure_action`	Users are returned to the model configuration page to make changes to line items. This action is automatically created when a Configurable Attribute Set is added to a sub-document.
Recalculate	`_calculate_price_action`	Users can re-calculate pricing data after a pricing field is altered. This action is automatically created when a Price Attribute Set is added to a sub-document.
Create [sub-document]	`_create_<sub_document_variable_name>`	Users can create another sub-document. The system automatically creates the Create action if there is another sub-document attached to the next main-document in order.



The system automatically deletes default, system-generated actions when their corresponding attribute or attribute set gets deleted.

Group Sequence Number

End users can view recommended items and child line items with sequence numbers that are counted based on the parent line item's sequence. **Example:** a parent line item can have a group sequence number of 1 and then the child items will have group sequence number of 1.1, 1.2, 1.3...etc. *This functionality is enabled when you add the* `_group_sequence_number` *attribute to the attribute list and the line item grid instead of* `_sequence_number`.



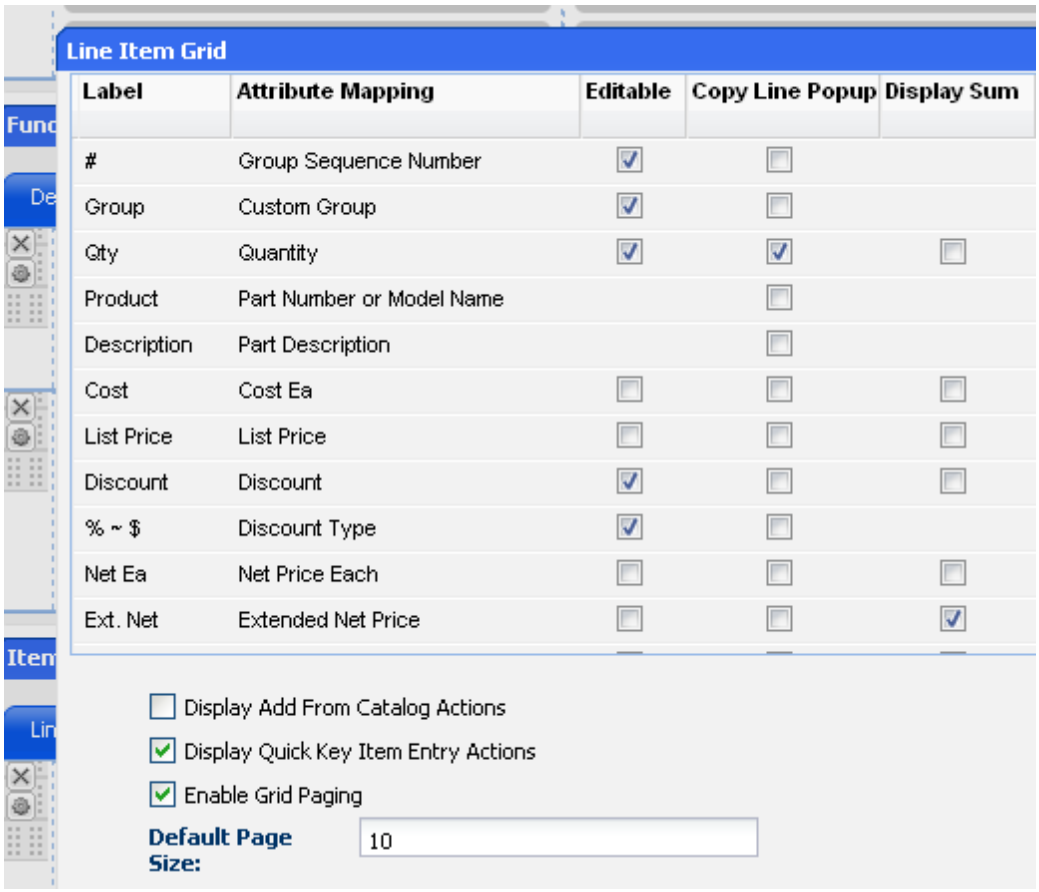
#	Group	Qty	Product	Description	Cost	List Price	Ext List	Discount	% ~ $	Ext Discount	Ovr. Net Ea	Net Ea
1			Mid Sized SUV									
1.1		1	F25X0040	F25X0040	$0.00	$743.75	$743.75	0.0	%	$0.00		$743.75
1.2		1	F25X	F25X	$0.00	$35,516.25	$35,516.25	0.0	%	$0.00		$35,516.25

Enabling Group Sequence Numbers

1. Verify there is a Group Sequence Number sub-document (line) attribute with the variable name `_group_sequence_number`.



2. Add the **Group Sequence Number** to the **Line item Grid** on the Main Document Layout Editor.



Line items are automatically re-sequenced if the user enters a value in the group sequence number field. However, if the user enters an invalid value, then the user's input is ignored.

Group Sequence Number functionality is independent of Sequence Number. *Group sequence number should not be used in conjunction with sequence number.*

Tabs

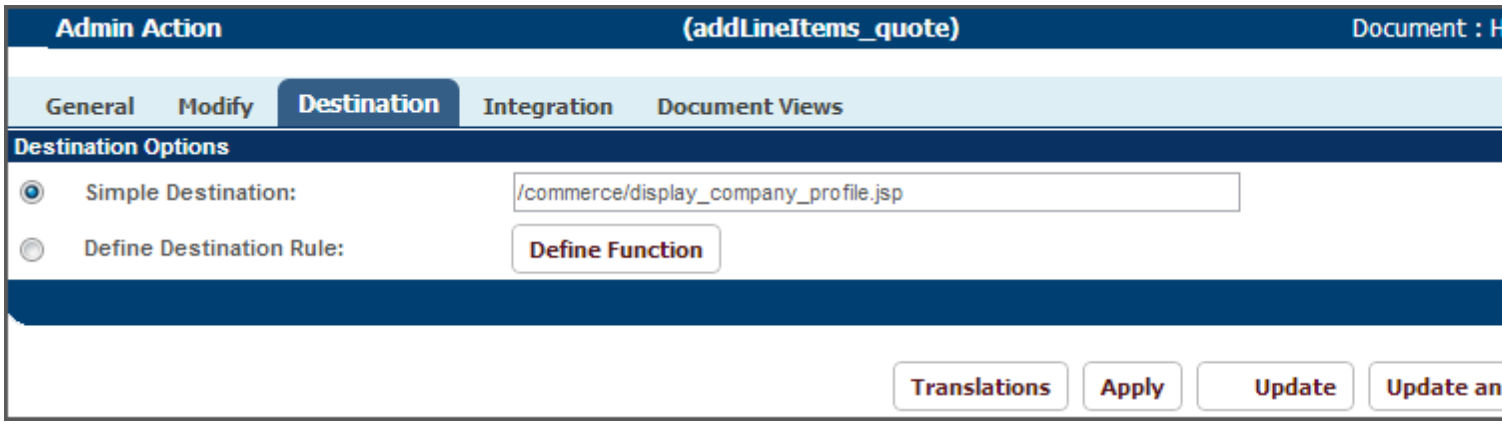
Destination Tab

You can create destination rules for the Modify and Add from Catalog type actions. *The Destination tab contains different options depending on the combination of the type of action and type of document:*

- Add from Catalog Action

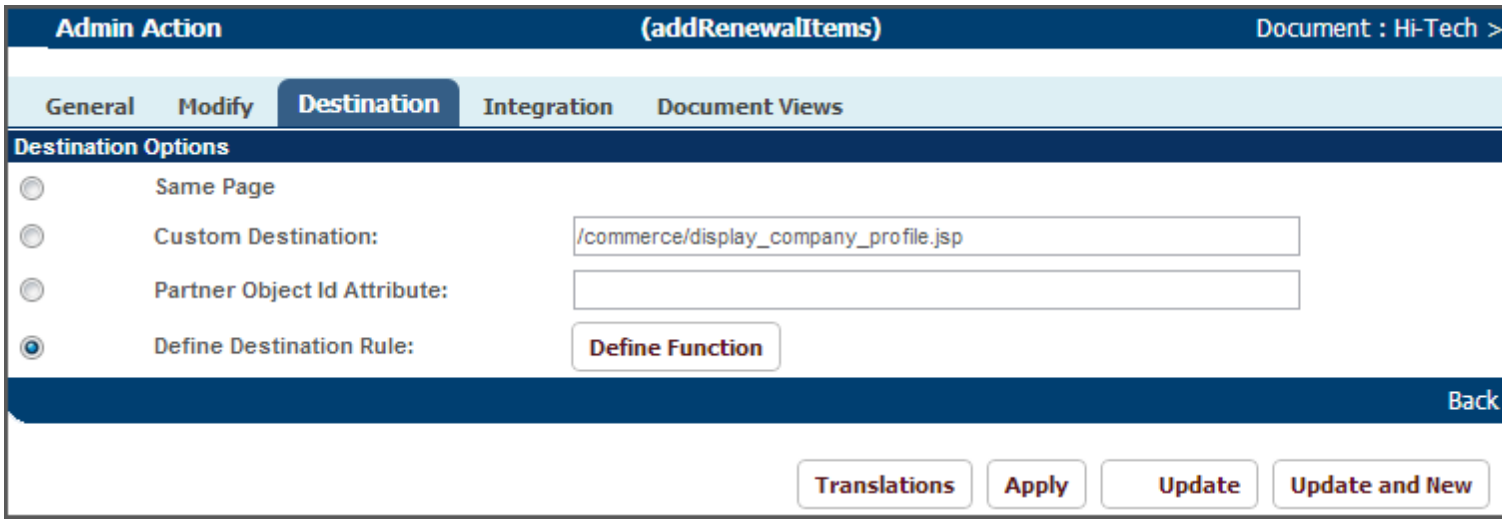
  This action supports both simple and advanced destinations. These destinations are specified with the use of URL. The user can pass information from commerce to the catalog by appending the data to the query string of the URL.

Destination	Description
Simple Destination	This option allows you to hard code a URL.
Define Destination Rule	This option allows you to write an advanced function that returns a string. The string that is returned specifies the destination URL. The inputs available to the Advanced Destination are system variables, main document and sub document attributes. *The Advanced Destination can also call commerce and utils library functions.*



- Modify Action

Options	Description
Same Page	Upon clicking the action, the user will remain on the current page. **Example:** on a modify action called 'Save' on the main document; the destination could be set to same page. Upon clicking this action, the user would remain on the current document.
Custom Destination	Upon clicking the action, the user will be taken to the URL specified in this field. *The valid value for this field is the relative path to any page on the BigMachines application.* **Example:** `/commerce/buyside/commerce_list.jsp` would be a valid value for the custom destination as the user will be taken to favorites page.
Partner Object ID Attribute	Upon clicking the action, the user will be punched-in to the appropriate object on the partner site. A valid value for this field would be the variable name of the attribute which stores the partner object Id information. **Example:** if a commerce attribute with variable name `partner_opportunity_id` contained the partner opportunity ID, this attribute variable name would be in the Partner Object Id attribute field.
Define Destination Rule	Upon clicking the **Define Function** action, the user will be taken to the URL returned by running a BML function. The valid value return value for this function is a string. Any URL whose relative path is a page on the BigMachines application would be a valid return value. **Example:** `/commerce/buyside/commerce_list.jsp` would be a valid value as the user will be taken to favorites page.



Rules-Based Destination using an Advanced Function

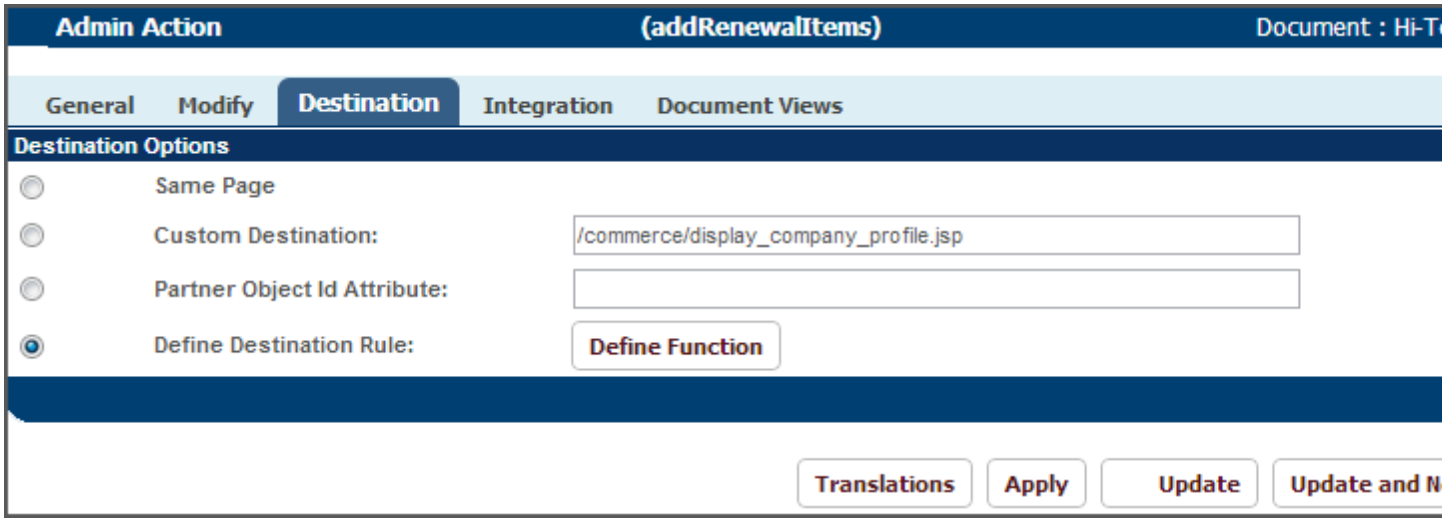
A destination script can be used to send an **Add from Catalog** type action to one page or another. This script appends the appropriate model name into a URL string, which is returned as the destination link. **Example:**

```
model_name="";
for a in quote_detail {
if( a._document_number == _system_current_document_number ) {
model_name=a._model_name;
}
}
if( model_name == "Big Pump" ) {
model_name = "Big%20Septic%20Tank";
}
if( model_name == "Big Septic Tank" ) {
```

```
model_name = "Big%20Pump";
}
return "acme.bigmachines.com/commerce/new_equipment/products/ model_con-
figs.jsp?segment=Mega%20Pumps&product_line=Gamera%20Pumps%20and%20Tanks&model=" + model_name;
```

Creating Destination Rules for a Modify Action

1. Select the radio button beside the desired **Destination Options**.



Destination Options	Description
Parent Page	Displays the parent page when the action is clicked.
Same Page	Displays the same page when the action is clicked.
Custom Destination	Display a custom destination. **Example:** `/commerce/display_company_profile.jsp`
Define Destination Rule	Displays a rules-based destination.

2. Click **Apply** to save changes. Click **Update** to save changes and return to the Actions List page. Click **Update and New** to save changes and create a new action. Click **Back** to return to the Actions List without saving changes.

3. If desired, select **Translations**: BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.

Creating Destination Rules for an Add from Catalog Action

1. Click the **Destination** tab.

2. Select the radio button beside the desired Destination Options.



Destination Option	Description
Simple Destination	Used to hard-code a specific destination page, either a page in the catalog or the Commerce List page. **Must enter the destination URL.** *Supports a single destination page.*
Define Destination Rule	Used to define rules that instruct the system where to take the user based on a condition. Supports different destinations. **Click Define Function.** *Can pass attribute values with a Reconfigure action.*

3. Select one of the following:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Apply:** saves changes and remains on the page.
   - **Update:** saves changes and returns to the Document List page.
   - **Back:** returns to the Document List page without saving changes.

Populating a Multi-Select Menu in Configuration

The values in all caps should be replaced accordingly. `CONFIG_ATT_VARIABLE_NAME=VALUE` should be used for the configuration attribute(s) that you want to set. To add multiple attributes, it should be delimited by "&".

```
http://yourcompany.bigmachines.com/commerce/new_equipment/products/model_configs.jps?
segment=SEGMENT_VARIABLE_NAME&product_line=PRODUCT_LINE_VARIABLE_NAME&model=MODEL_VARIABLE_NAME&CONFIG_ATT_
VARIABLE_NAME = VALUE&variable_name_punchin=true
```

To set a multi-select menu, the values in `VALUE` should be separated by a `~`.

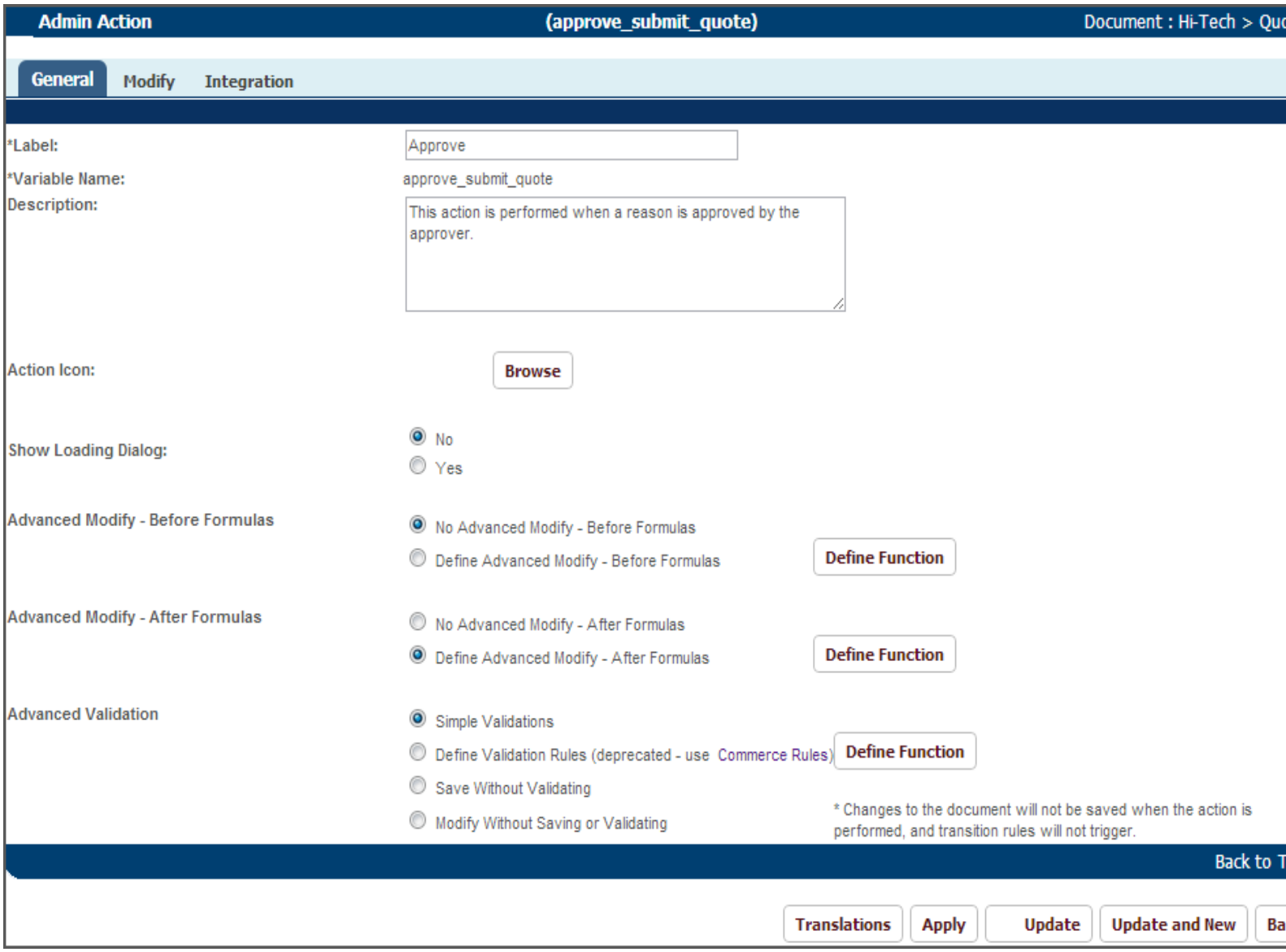
Example: `CONFIG_ATT_VARIABLE_NAME = VALUE1~VALUE2~VALUE3~VALUE4`

Document Views Tab

The Document Views tab allows the fullaccessuser to control the display of an action to users in different steps of the transaction. The fullaccessuser can make the action active or inactive in a certain step to a particular participant profile.

General Tab

The General tab is the only tab that is common to all action types. It holds the action information such as Label, Variable Name, Modifications and Validations.



Element	Option	Description
Label		Text label that is displayed on the action button.
Variable Name		Unique required field identifying the action in rules. *Cannot be modified once it's been saved.*
Description		Description of the action.
Action Icon		Icon that is displayed on the transaction button.
Show Load-	• Yes	This option turns on or off a loading dialog on the commerce side when the user clicks on the action. The image

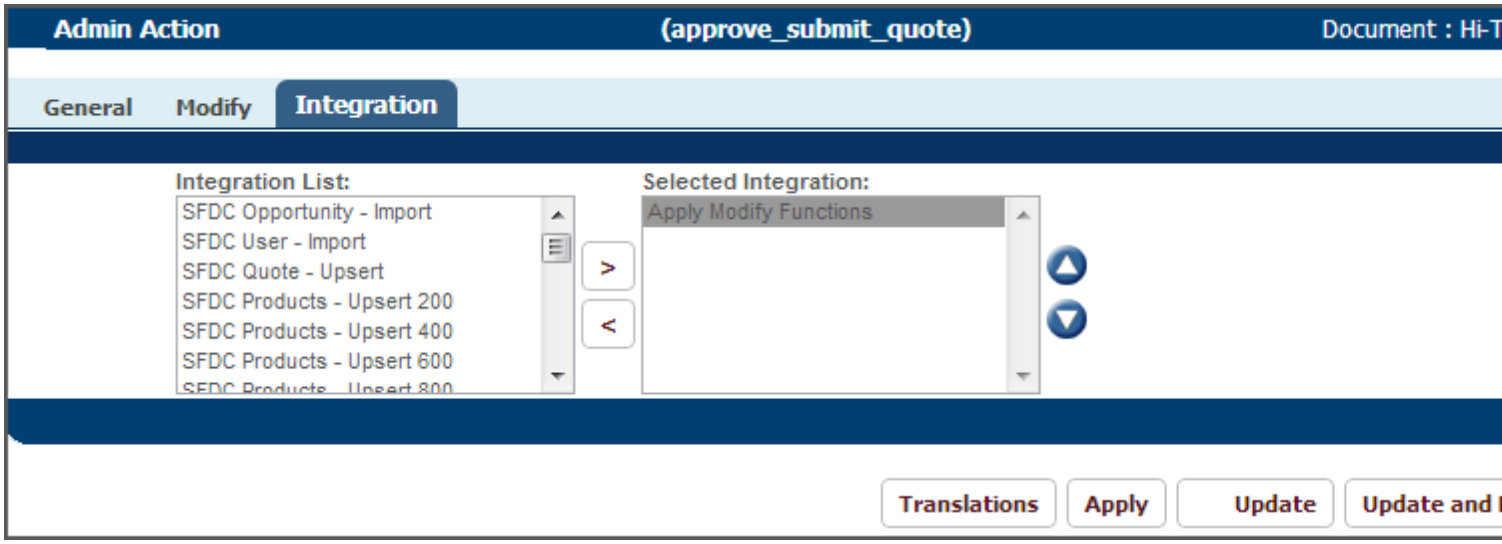
Element	Option	Description
ing Dialog	• No	and text for the loading dialog can be customized in the site settings.
Layout Path		Button that opens up the Layout Editor located on the main document.
Advanced Modify- Before For-mulas	• No Adv Modify-Before Formulas • Define Adv Modify-Before Formulas	Advanced modification refers to an action's capability to modify attribute values in the current document using an advanced function when the user clicks on the action. When this option is selected, the advanced modification will be run before the formulas created in Formula Management are used.
Advanced Modify- After Formulas	• No Adv Modify-After Formulas • Define Adv Modify-After Formulas	Advanced modification refers to an action's capability to modify attribute values in the current document using an advanced function when the user clicks on the action. When this option is selected, the advanced modification will be run after the formulas created in Formula Management are used. *Advanced modification is run at the end and therefore can override the value set by an attribute's modify tab.*
Advanced Val-idation	• Simple Validations • Define Validation Rules- Deprecated • Save Without Val-idating • Modify Without Sav-ing or Validating	Advanced validation refers to an action's capability to validate the attribute values in the context of the current doc-ument and its associated parent/ child, when the user clicks on the action. **Example:** user can validate an attrib-ute on a sub-document using an advanced validation function on the main document and vice-versa. Basic data type validation is always done for all attribute types. *The system supports three types of validation settings for document actions:* 1. Simple 2. Advanced 3. No Validation **Simple Validation:** System checks attribute values using individual attribute validations. *Default validation set-ting for document attributes.* Simple validations can include checks for numeric values, numbers in a range, required attribute values, or any other type of check that can be set for an individual attribute. **Advanced Validation:** System performs cross-validations in addition to simple validations. Cross-validations between attribute values can be created using the Script Editor. Example: create validation at the document level that prohibits discounts in excess of 20% when a purchase price is under $1000. *This type of validation can also be created as a constraint configuration rule.* **Deprecated- Use Commerce Rules.** **Save Without Validating:** Turns off all checks on attribute values when the action is being performed and saves the document data. This option guarantees that an action will succeed and is useful for actions set up on timers, such as an expiration action. **Modify with Saving or Validating:** This option is similar to the "Avoid All Validation (Save)" action, only doc-ument data is not saved when the action is performed. This validation method might be selected for a modify type action set up to copy data from one set of fields to another.
Maximum Number of Copies		**Only available for Copy Line Items action.** Specifies the maximum number of copies a user can make. If you choose a number greater than one, a popup appears on the screen and the user can input their desired number of copies. The fullaccessuser can map the attributes that are displayed on the popup by selecting "Display in popup for Copy Line Item" in the Admin Column page for Line Item Columns.
Default to Quantity		**Only available for Copy Line Items action.** If the fullaccessuser selects this option, then the number of copies field in the Copy Line Item pop-up is pre-pop-ulated with the quantity of the selected line item(s). If the quantity of the selected line items is greater than the max-imum number of copies specified by the fullaccessuser, then the Number of Copies field is pre-populated with the maximum number of copies.

Defining an advanced modification allows you to change the values of multiple attributes at the same time when a particular action is clicked. **Example:** it will return values from a Commerce Library Pricing Function, when a user clicks **Save**, **Submit**, or **Update Line Items**. This allows you to create the price logic once and run it on multiple actions.

Defining an advanced validation allows you to display an error message if the user has clicked an action, but has entered something incorrectly. **Example:** you could write a BML script that has an error message reading- "Line Item 2 exceeds the maximum allowable discount."- based on your particular parameters.

Integration Tab

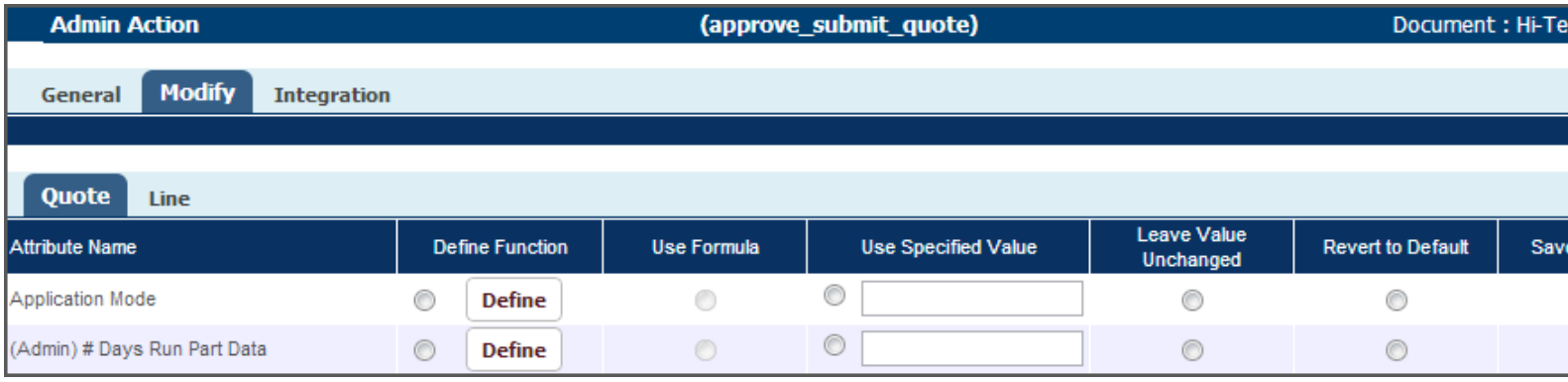
The Integration tab (displayed only if the site is integrated to a partner) determines the order in which data is exchanged between BigMachines and a partner application. The order in which the integration XSLs are specified in this tab determines the order in which data is either sent to or received from the partner.



The Apply Modify value in this tab cannot be removed. All integrations can be ordered around it. Also, no integration can be performed without performing a Modify tab operation.

Modify Tab

The Modify tab is available for all 'Modify' type actions (system-generated or user-defined). This tab controls the manner in which a transaction gets its data; it allows you to manipulate the values in the attributes by specifying behavior in this tab. The Modify Tab contains a list of all attributes available in the current tab with five options for each attribute.



Using the Modify Tab to Set Your Behavior

1. Click the **Modify** tab.
2. Select one of the following actions for each listed **Attribute Name:**

Action	Description
Define Function	This option implies that the user must define a function that calculates the attribute value. Click **Define** after selecting the radio button. Use the Script Editor to create a function that sets the attribute value. You can use this option to determine current system variables, such as the currency conversion rate or system date.
Use Specified Value	This option implies that the attribute will get its value specified in the corresponding field. Enter a value in the value field next to the radio button.
Leave Value Unchanged	This is the default option. It implies that the attribute will retain its current value, whether its a user input or system defined.
Revert to Default	This option implies that the attribute value will be reset to its default value.
Save from Form	This option implies that the attribute will always retrieve its value from the user input on the commerce side.

3. Select one of the following:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Apply:** saves changes and remains on the page.
   - **Update:** saves changes and returns to the Document List page.
   - **Update and New:** saves changes and creates a new action.
   - **Back:** returns to the Document List page without saving changes.



When setting modify rules for the **Autofill** type document action, the options set on the **Modify** tab have a higher precedence than mapping rules set on the **Mapping** tab. To force the system to use mapping rules for auto-filling fields instead of modify rules, use the **Leave Value Unchanged** setting. If this setting is used and mapping rules are not available for a field, then the system populates the field using the attribute's default value.

You can configure all listed attributes the same way by choosing an option and selecting the **Select All** radio button at the bottom of the column.

Special Tabs

- **XSL Views:** These action types must be used in conjunction with an XSL template file to produce a commerce side output. Such actions can be associated with one or more XSL views. If multiple XSL views are available, users can select an XSL template from the ones available. The XSL views option applies to print, email and display history actions; it enables you to support custom forms and page displays using XSL and XSL FO. *XSL Views are usually associated with the following action types:*
    - Print
    - Email
    - Display History
    - Export Attachment
- **Email Fields:** Email actions contain a special tab called Email Fields. Email field rules instruct the system to auto-populate fields in the email pop-up window. With the fields populated by the system, users can send out emails quickly. *Email field rules apply to the email document action only.*
- **Mappings:** Available only for actions related to the accounts section. Account fields and commerce attributes are mapped in this tab. When the user selects a customer account ID and clicks on a relevant action, the mapping specified in the respective action will populate the commerce attributes with data from the corresponding accounts fields. Mapping rules are used to link data from one document to another in the same process, or from the Account database to a commerce document. **Example:** use mapping rules to link Account customer records to a commerce document's 'Bill To/Ship To' fields.



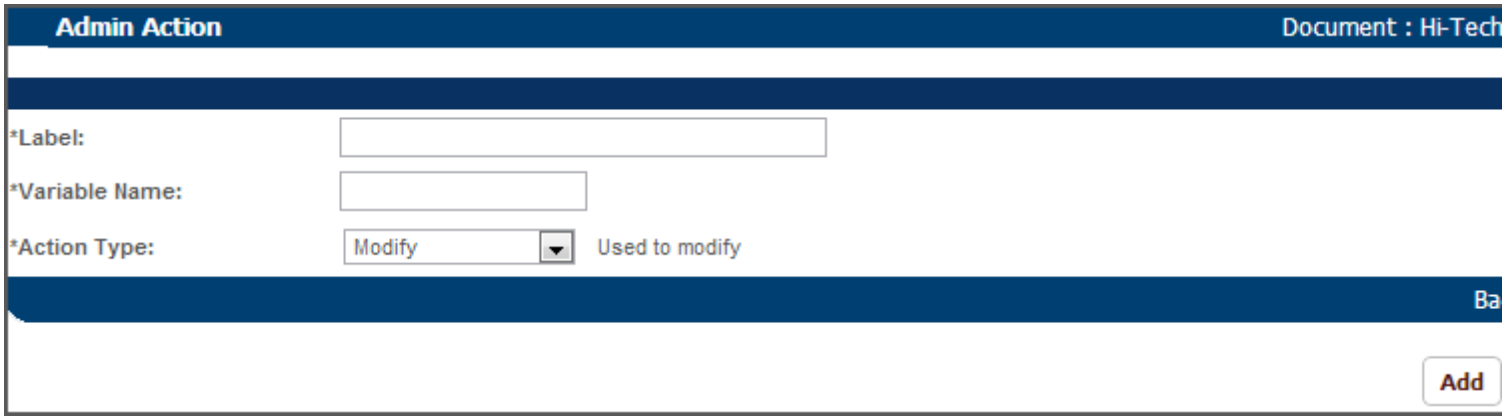
Mapping rules have a higher precedence than an attribute's default value, however, they have a lower precedence than Modify rules. To force the system to use mapping rules when a user performs an **Autofill** or **Select Address** action, use the **Leave Value Unchanged** setting on the Modify tab.

When a user performs an **Autofill**, **Browse** or **Populate** action, the system uses the mapping set up for the **Autofill** action. When a user performs a **Select Address** action, the system uses the mapping set up for the **Select Address** action.

ADMINISTRATION

Adding Document Actions

1. Click **Add** at the bottom of the **Actions List** page.
2. Enter a **Name**. *This field will be used to link back to edit the action.*
3. Enter a unique **Variable Name**. *Do not use any special characters on this field.*



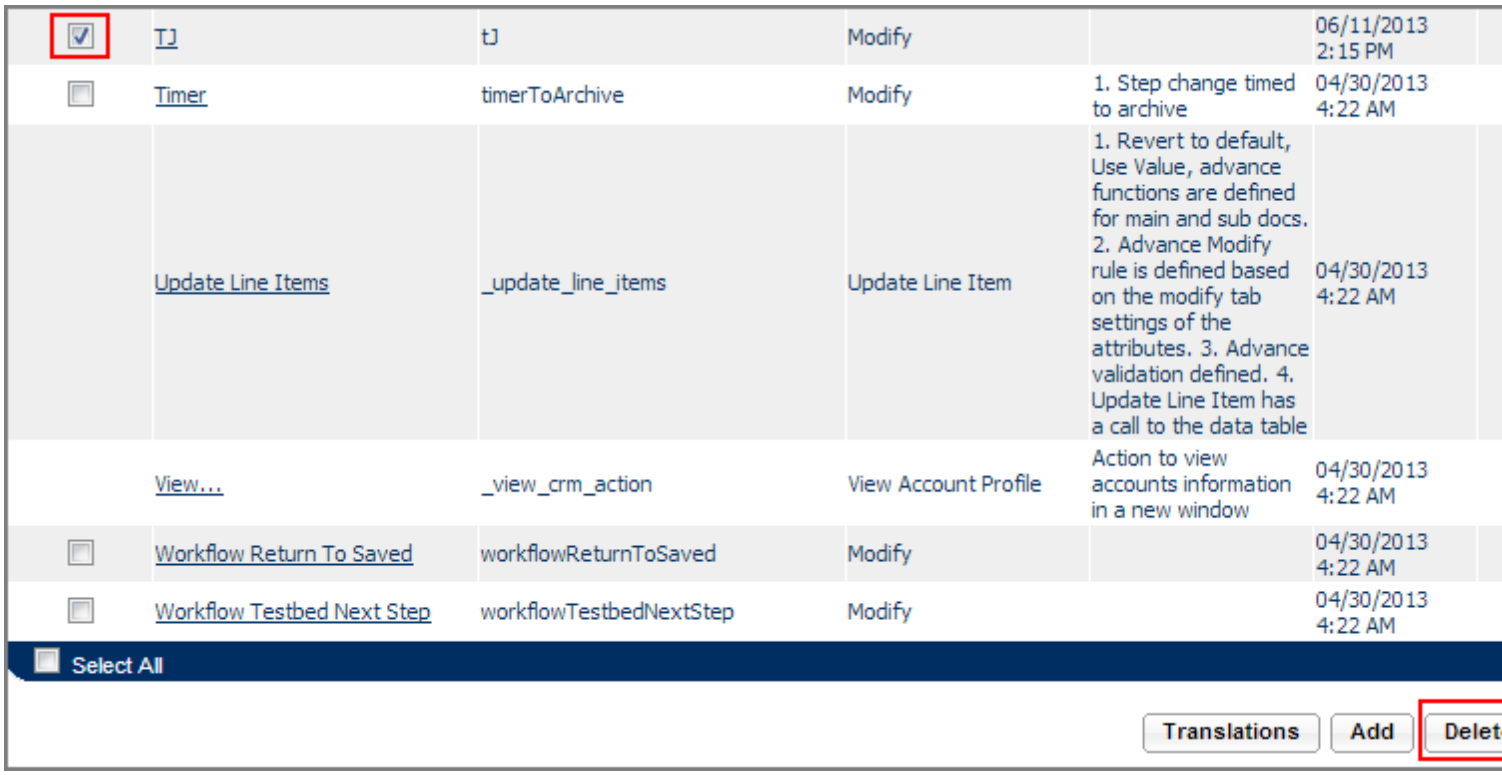
4. Select your desired **Action Type** from the drop-down menu.
5. Click **Add** to save changes and open Admin Action editor or click **Cancel** to return to Document List without saving changes.
6. Edit the tabs available. *Refer to the Tabs section above.*
7. Select one of the following:
   - **Translations:** BigMachines content and product templates can be translated into the following languages: French, German, and Spanish.
   - **Apply:** saves changes and remains on the page.
   - **Update:** saves changes and returns to the Document List page.
   - **Update and New:** saves changes and creates a new action.
   - **Back:** returns to the Document List page without saving changes.

The new action appears in alphabetical order on the **Actions List** page.

Variable names cannot be edited after they are created.

Deleting Document Actions

1. Mark the checkbox of the desired under the **Select** column within the **Actions List** page.



2. Click **Delete**.



You can delete all your actions by checking the **Select All** box and then clicking **Delete**.



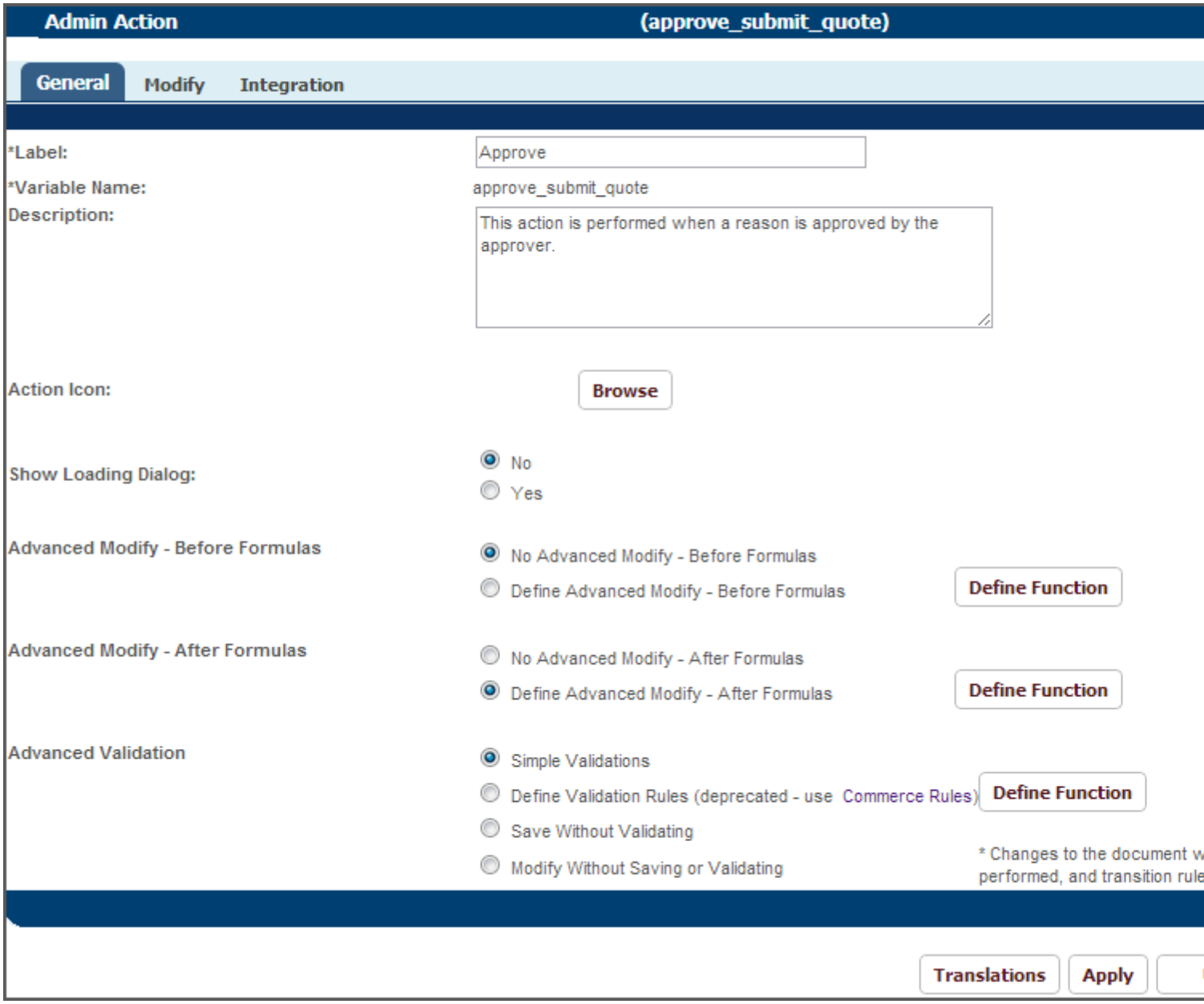
It is not possible to delete default, system-generated actions. *For default actions created in association with document attributes, the system deletes these actions automatically when the corresponding attribute is deleted.*



Use caution when deleting a document action because it is not possible to recover the deleted data. Upon a deletion, all associated rules within the document action are also deleted.

Editing Document Actions

1. Modify the **Label**. *The Variable Name field will appear in read-only format in edit mode.*
2. Modify the **Description**, if required. *The description is for your reference and appears on the Actions for Document page.*



3. Edit the desired rules associated with the action.

Edits are not visible until the process is deployed (or redeployed). All actions besides system defined actions can be deleted even after a process has been deployed.

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). *This eliminates the need for conditionals to segregate logic.*
    - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
    - The Reconfigure Inbound action only contains what is specific to it.
    - This allows administrators to execute a formula per attribute on:
        - Outbound paths from a Commerce Transaction. *Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.*
        - Inbound paths to a Commerce Transaction. *Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.*
    - Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES



In order for any action to appear on a commerce document, it must be included in the document view.



Document Actions can be created, deleted, and edited one document at a time.



User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.

RELATED TOPICS

USE CASE - QUOTE VERSIONING

OVERVIEW

You can enable Quote Versioning by adding a reference to the original quote number, transaction ID and a link as Transaction fields populate them as part of the copy/clone Transaction process. When troubleshooting quote-specific commerce issues, or when an administrator would like to see why a quote was configured in a particular way, it is useful to be able to quickly determine whether a quote was copied from another transaction and reach the source transaction.

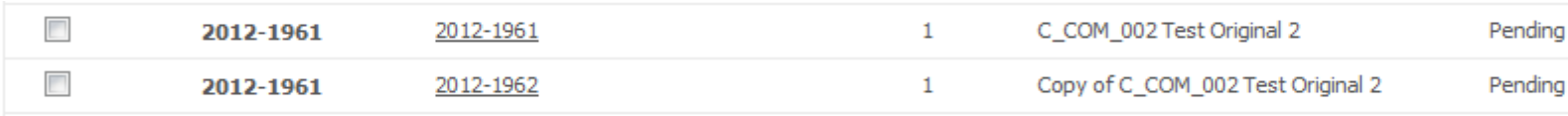
The sequence of events on the standard Copy Quote action will set the values for the five new commerce attributes you will create.

Prerequisites

Before using this tutorial, you must:

- Understand how to create commerce attributes
- Be familiar with BML
- Be able to navigate a commerce process
- Be a Blue Belt Level Administrator

Transaction Manager with Quote Versioning



	2012-1961	2012-1961	1	C_COM_002 Test Original 2	Pending
	2012-1961	2012-1962	1	Copy of C_COM_002 Test Original 2	Pending

ADMINISTRATION

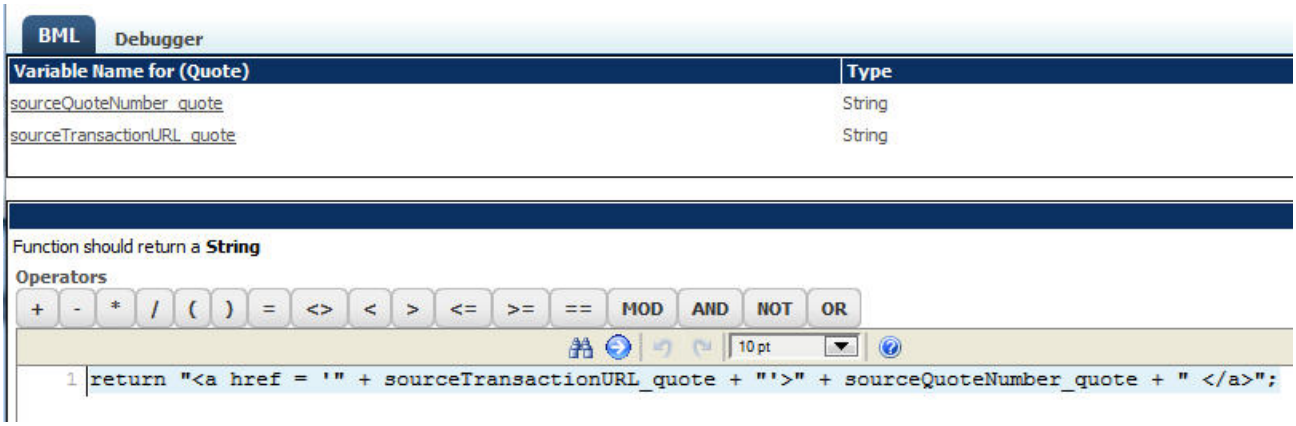
Enable Quote Versioning

1. Create new main-document attributes. Click here for a complete list.

Attribute	Attribute Type	Variable Name	Description
Is Clone?	Boolean	isClone_quote	Will be TRUE if the transaction started as a copy of another.
Source Quote #	Text	sourceQuoteNumber_ quote	Quote Number of the source transaction.
Source Transaction ID	Text	sourceTransactionID_ quote	Transaction ID of the source transaction.
Source Transaction URL	Text	sourceTransactionURL_ quote	URL of the source transaction, which is derived from the linkToDocu-ment_quote.
Source Quote	Read-only text or HTML	sourceQuoteLink	Link to source quote. **See Step 2.

2. Click the **Default tab** for the **Source Quote** attribute.

3. Select **Function** and click the **Define** button.
4. Select the Quote variables: **sourceTransactionURL_quote** and **sourceQuoteNumber_quote**.
5. In the script definition area, enter the following, then click Save & Close:



6. Re-order the attributes so that they appear in the following order:
   - sourceQuote Number_quote
   - quoteNumber_quote
   - sourceTransactionID_quote
   - transactionID_quote
   - sourceTransactionURL_quote
   - linkToDocument_quote



This is done to ensure that the value that is saved is the value from the SOURCE transaction and not the re-initialized clone.

7. Navigate back to the Commerce Process Level > Process Actions > List > Copy.
8. Click the **Initialization tab**. *On this tab, you will need to map the new attributes to the attribute values in the original transaction.*
9. Select Define and click the Define button next to **Is Clone?**.
10. Navigate to the script definition area and enter the following, then click Save & Close:

```
return TRUE;
```

11. Select Define and click the Define button next to **Source Quote #**.
12. Click the Variable Name for (Quote) tab, select **quoteNumber_quote** from the list, then click **Next**.
13. Navigate to the script definition area and enter the following, then click Save & Close:

```
return quoteNumber_quote;
```

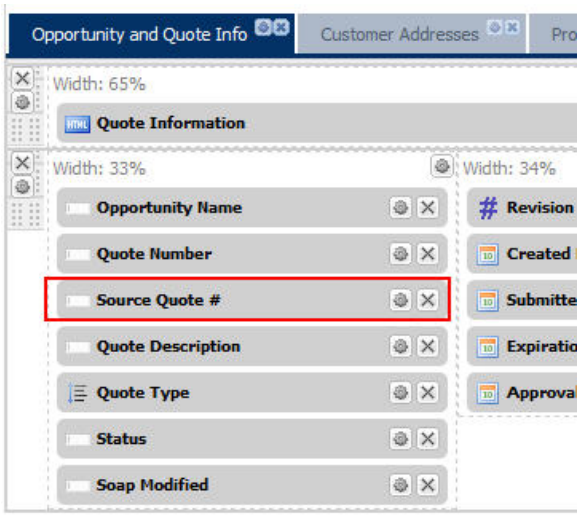
14. Select Define and click the Define button next to **Source Transaction ID**.
15. Click the Variable Name for (Quote) tab, select **transactionID_quote** from the list, then click **Next**.
16. Navigate to the script definition area and enter the following, then click Save & Close:

```
return transactionID_quote;
```

17. Select Define and click the Define button next to **Source Transaction URL**.
18. Click the Variable Name for (Quote) tab, select **linkToDocument_quote** from the list, then click **Next**.
19. Navigate to the script definition area and enter the following, then click Save & Close:

```
return linkToDocument_quote;
```

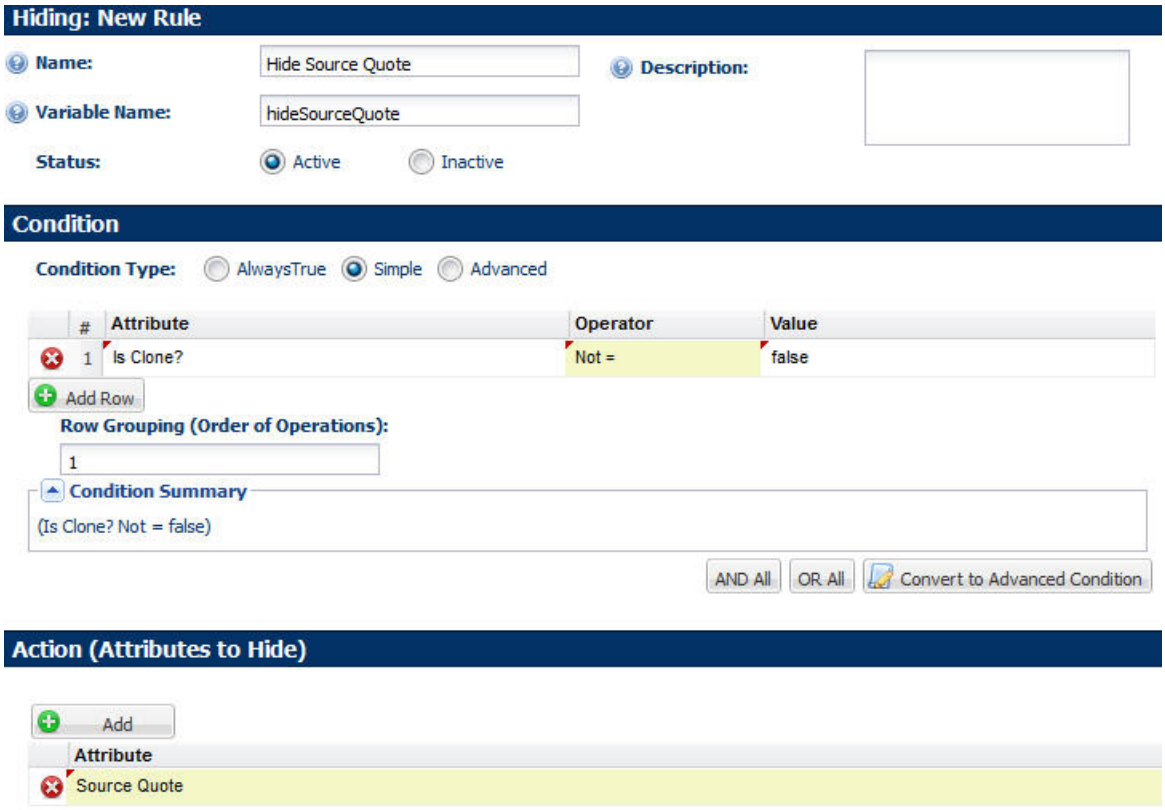
20. Navigate back to the commerce process level, then add the field(s) to the quote layout.



Optional Steps

You may want to know if a particular quote is a copy and if it's not, hide the Source Link attribute.

1. Create a quote-level commerce hiding rule called ‘Hide Source Quote’.



RELATED TOPICS

USE CASE - DATA DRIVEN DROP-DOWN MENU

OVERVIEW

What Does It Do?

Dynamic data driven drop-down menus are not native within BigMachines. An HTML attribute, along with a hidden attribute to store the selection, must be used to accomplish this behavior.

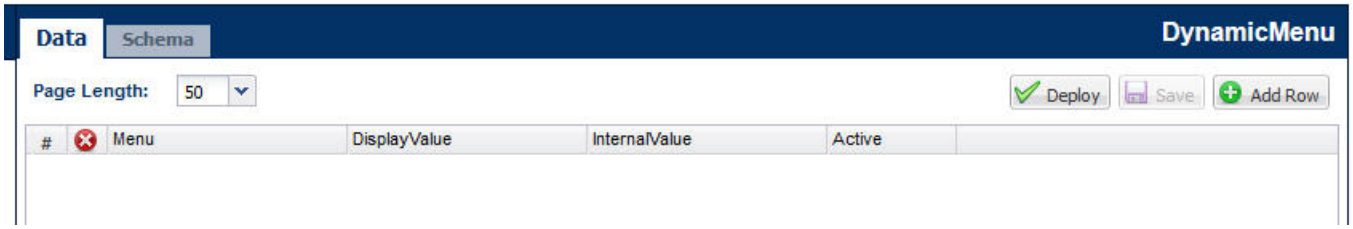
How Does It Work?

You will create Util function that will read all of the values from the database and build out the <select> HTML tag. It will automatically build out the JavaScript onChange event so that the selected value is pushed as clear text into the second attribute.

ADMINISTRATION

Step 1: Create Data Table: DynamicMenu

1. Navigate to Data Tables from Admin Home Page, Quick Links or Navigate Menus.
2. Click **Add** to add the table.
3. Click the **Schema Tab** and create the following **string** columns:
    - **Menu (String):** The list name. All rows for the same list need the same keyword. This needs to match the MenuName passed to the Util Function.
    - **DisplayValue(String):** The value to display in the screen.
    - **InternalValue(String):** The value to push into the hidden attribute. Many times this matches the DisplayValue, but doesn't have to.
    - **Active(Integer)**

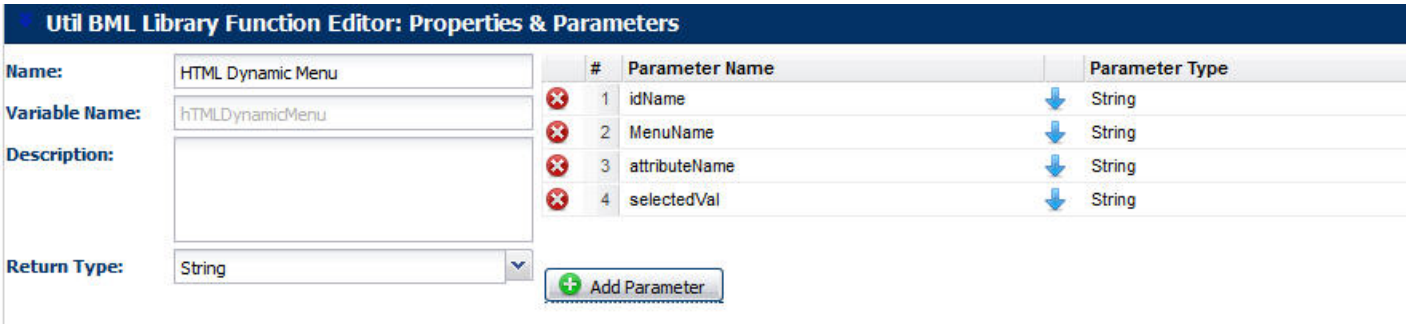


4. Click **Deploy**.

Step 2: Create Util Function

1. Navigate to **BML Util Libraries**.
2. Click **Add**.
3. Name the function: **HTML Dynamic Menu (HTMLDynamicMenu)**.

4. Choose **String** as the Return Type.
5. Create these four string parameters:
    - **MenuName :** Links to the data set within a data table.
    - **idName:** The HTML id for the HTML widget.
    - **selectedVal:** Menu Item that should be preselected.
    - **attributeName:** The attribute name that should have the selected value copied to it. Generally this is a hidden attribute, but doesn't necessarily have to be.



6. Click here for Util Function Script.

```
//HTML Dynamic Menu Library Function

ret = "";

ret = ret + "<select id=\"" + idName + "\" onchange='$(\"input[name=" + attributeName + "]\").val($(this).val
());' style=\"width: 100%\">";


//Create the blank option string.

ret = ret + "<option value=\"\"></option>";


//Initialize hasCurrentValue flag

hasCurrentValue = false;


//Query the DynamicMenu table, grabbing all other options

menuOptions = bmql("SELECT DisplayValue, InternalValue from DynamicMenu WHERE Menu = $menuName AND (Active = 1 OR
InternalValue = $selectedVal)");

for option in menuOptions {

dispVal = get(option, "DisplayValue");

intVal = get(option, "InternalValue");
```

```
//Check to see if the option is the current selected value. If so, make option the default selected value.
//Also, set hasCurrentValue to true;
if(intVal == selectedVal){
ret = ret + "<option value=\"" + intVal + "\" selected=\"selected\">" + dispVal + "</option>";
hasCurrentValue = true;
}

//Create a normal option otherwise
else {
ret = ret + "<option value=\"" + intVal + "\">" + dispVal + "</option>";
}
}

//If current value not in table, create special entry for it.
if(hasCurrentValue == false AND selectedVal <> ""){
ret = ret + "<option value=\"" + selectedVal + "\" selected=\"selected\">" + selectedVal + "</option>";
}

//Close out the menu tags
ret = ret + "</select>";
return ret;
```

7. Click **Add**.
8. Select **HTML Dynamic Menu** function and click **Deploy**.

Step 3: Create Commerce Attributes

1. Navigate to Quote Level (main-document) for a Commerce Process > Attributes> List.
2. Click **Add**.
3. Name the attribute: **Create Menu (createMenu)**.
4. Choose **Ready-only text or HTML** as the Attribute Type.
5. Click **Add**.
6. Click the **Default** tab.
7. Select the **Function** radio button and click **Define Function**.
8. Copy and paste the following code:

```
return util.HTMLDynamicMenu("Operations", "operationsDC1", operationsApprovalDC1_quote, "operationsApprovalDC1_
quote");
```

9. Click **Save and Close**.
10. Click **Update and New**.
11. Name the new attribute: **Real Attribute (realAttribute)**.
12. Choose **Text** as the Attribute Type.
    - Generally hidden while the HTML attribute is displayed when Security or Process dictates Read-Only view.
    - Set by JavaScript with <select> onChange.

Step 4: Populate DynamicMenu Data Table

1. Populate the DynamicMenu data table with the appropriate items.
2. All items need the same MenuName that is passed into the HTML Dynamic Menu Util.

NOTES



This solution uses custom JavaScript. This should only be adopted if you have a strong technical team.

RELATED TOPICS

Article Submitted By: Savvis

MIGRATION MANAGEMENT

OVERVIEW

Migration Management has been improved to allow you to easily pull changes from one BigMachines environment to another. To do this, BigMachines features a migration center that allows you to migrate the Util Library, Product Definition, Product Catalog, Attributes and Rules within Configuration, Commerce Process and Documents. In previous versions, test to production was done by downloading XML from the test site and uploading them on to the production site. Much of this Test to Production process has been replaced with Migration. Migration works by pulling data from the source site (test) to the environment you are currently in (production).

Highlights

- One-Touch Migration for commerce processes, documents, product family, catalog, configuration and util libraries
- View differences between the source and target sites prior to migrating
- View detailed differences of code change prior to migrating

---

Migration Basics

When migrating, changes are pulled from the source site to the target site. Commonly, the source site is referred to as 'test' and the target site as 'production'. Since migration works by pulling data fro the source site, the target site is the one that you are logged into and where you're accessing the Migration Center.

Only deployed data can be migrated. Additionally, detailed differences show the differences between the deployed data in both sites. On the target site, migrated components are automatically deployed. For example, if a commerce process is migrated from source to target, the process will be deployed on the target site after the migration is finished.

A component created manually or through bulk services on the source and target instances cannot be considered equivalent migration, even if they have the same variable name. For example, a commerce process on the source site with the variable name "myProcess" is not equivalent to a commerce process created manually on the target site with the same variable name. "myProcess" should have been created on the target site, through migration, for it to be equivalent to the process with the same variable name on the source site. Attempting a migration in this case will throw an error. We advise you to delete the manually created component on the target instance and then perform a migration.

ADMINISTRATION

The Migration Management Center allows you to migrate individual elements of your application in one central location.



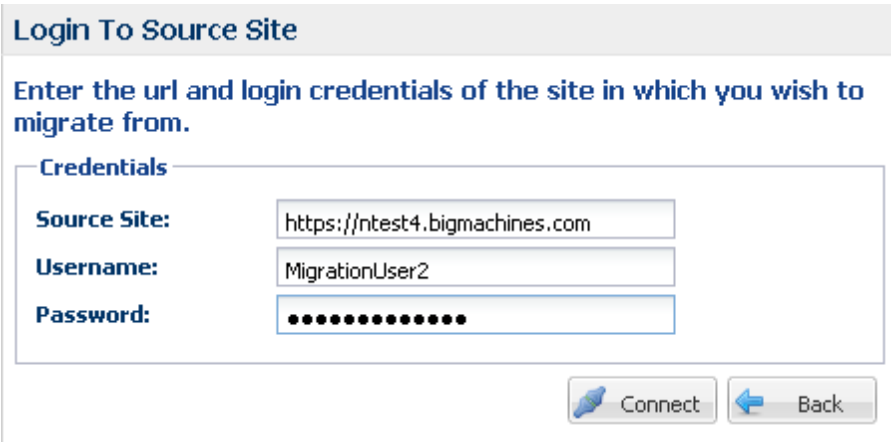
Table of Migration Center Elements

Element	Description
Migrate Button	Click Migrate after you have selected the module(s) that you would like to migrate from a specified source.
Status Button	Click Status to check the migration status. This is also where you will find a link to the summary log that provides more detailed information on successful or failed migrations.
Refresh Button	Click Refresh to re-evaluate the differences between your current environment and the source environment you are migrating from. This will help you determine what modules to migrate. Note: Each refresh calls the source site using system resources to process the data. Clicking refresh many times in a row could slow the loading of the left pane.
Source Site	Displays the URL of the environment you are migrating from.
Util Library	Contains a list of the Util Library functions that can be individually migrated from one environment to another.
Product Definition	Contains a list of the product families that can be individually migrated from one environment to another.
Catalog	Contains a list of the Product Lines and their subsequent Models that can be individually migrated from one environment to another. Note: This only migrates the catalog hierarchy.
Configuration	Contains a list of all the rules and attributes associated with each product family. Included are configurable attributes, rules, configuration and search flows, integration XSL, and stylesheets.
Commerce	Contains a list of all the Commerce Processes that can be individually migrated from one environment to another.
Document Engine	Contains a list of documents, generated by the Document Engine and sorted by Commerce Process, that can be individually migrated from one environment to another.

Logging into the Migration Center

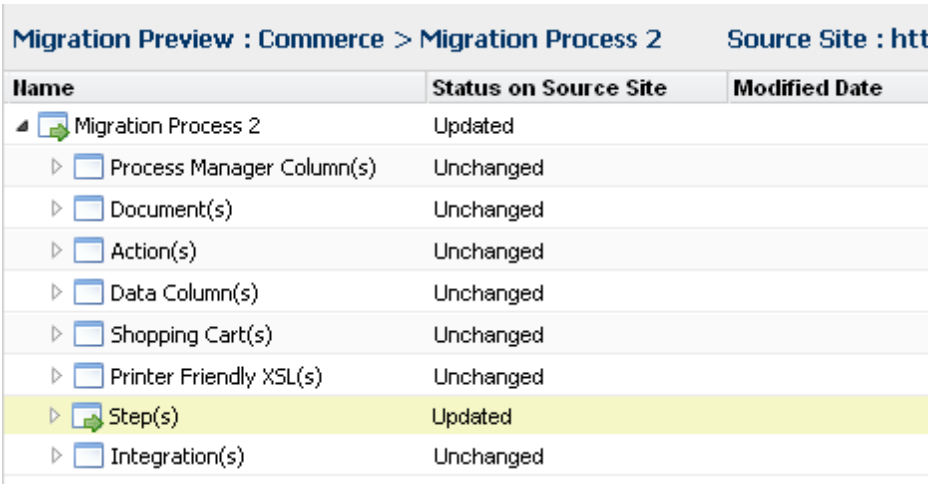
Upon accessing the Migration Center on the target site, you'll be asked to log into the source site. Please note: you can set the source site for a particular BigMachines instance by default. The user must login as a service user or as a host company superuser of the source site. Before you can access the migration management center, you first need to log in to your test environment

1. Navigate to Admin Platform > Utilties > Migration
2. Enter your credentials for the source site (commonly the test site) in the pop-up window.



3. Click **Connect**. A loading box will appear while the connection is being established, refreshing the data from the source site and calculating any differences between the two environment.

4. Upon successful connection, you'll see the Migration Center. In the right panel of the migration repository, you'll see a Migration Preview. If you click on an individual element with a module (or folder), you will see what it contains.



5. In the image above, there are three columns displayed: Name, Status on Source Site and Modified Date. Icons with green arrows indicate that this level may be expanded to display child elements.

Element	Description
Name	These are the names of the elements within the module. You can click the arrow to expand the list to see child elements.
Status on Source Site	This displays the status of each element on the source site when compared to the same element on the current site. 'Updated' incidcates that this element differs between the source site and current site. 'Unchanged' means that the source stie and current site are the same for this element.
Modified	This reflects the date in which the element was modified, inserted or deleted in the source environment.



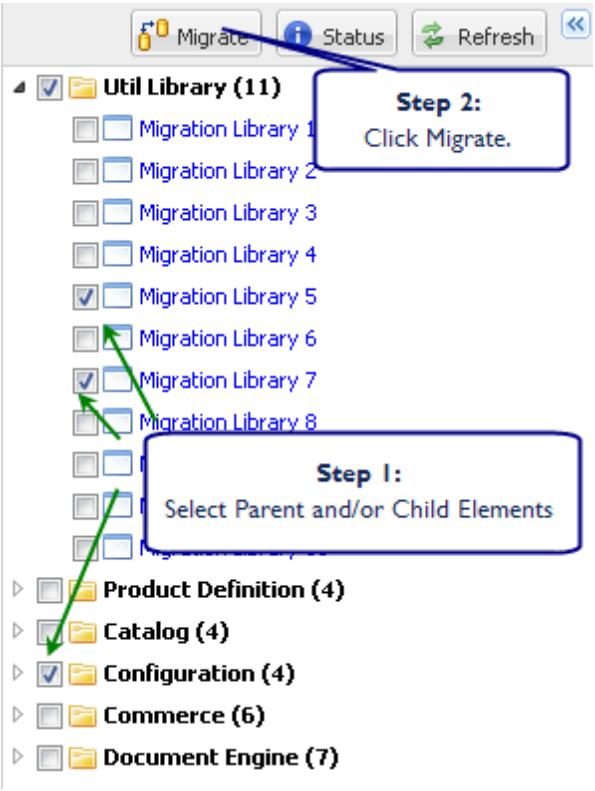
Click on the column header to see filtering options. Each column header allows you to select or deselect the columns you'd like to see. You can also perform an ascending or descending sort Status on Source Site and Modified Date columns

---

Migrating from the Source Environment

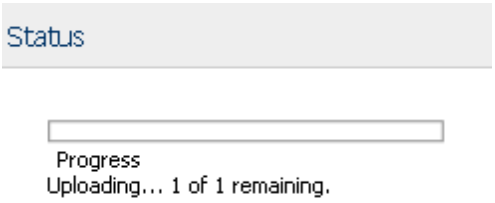
You can select a folder(module) in its entirety, or you can select individual elements you'd like to pull from the source environment.

1. Select Parent and/or Child Elements.



2. Click **Migrate**.

3. A status bar will appear in the bottom of the left panel. For migration of many items, the text provides a real-time indicator of the progress.



A dialog window attached to migration button allows the user to enter a migration description which is then displayed on the Migration Status page for reference.



Source and target sites must be on the same version in order to migrate. When upgrading to a new BigMachines version, make sure that the target and source sites are database copies of one another. First upgrade your test site (source), then upgrade production (target).

Commerce Migration Process

Standard Navigation Path: Administration Home Page > Commerce Processes > Process Definition > Migration

This functionality allows you to map old commerce processes to new commerce processes by selecting a process from each list, and clicking Add. After mapping commerce processes, set up document and workflow step mapping by choosing a mapped commerce process set, and clicking Document Mappings or Step Mappings. After all mappings have been set up, you can migrate commerce process data.



The migration feature is turned on based on a commerce global setting under Administration Home Page > Global System Settings and Utilities > Commerce.

NOTES



Nominate one fullaccessuser to perform migrations.

Remember that items not available in the Migration Center can be migrating by using Bulk Upload/Download. Some examples include: Accounts, Commerce Transactions, Parts and the File Manager.

Establish a process for migration – It is strongly recommended that migrations be performed one-way. For example, always migrate from the test to the production instance and not vice versa. The System fullaccessuser can set it up such that the source site name is auto-populated in the Migration page of the target site.

Entities that are migrated should not be created manually or through bulk upload/download on the target site. Manually created entities, or those added through bulk services, are not considered equivalent for migration, even if they share the same variable name.



Up to 10 snapshots are stored at once. For more information, see Rollback & Snapshots.

Rollbacks will not affect transactions; a Rollback does not delete Steps.

Data Columns can be deleted through Migration. *Deletion of a Data Column via Migration will have the same effect on Transaction Manager Views and Reports on the target site as if it were deleted in a Rollback.*

Only users with access to perform Migrations can also perform Rollbacks.

Status emails are sent to the Superuser after Rollback is complete.

Users will experience an increase in total Migration time due to the addition of the snapshot that must be taken to facilitate the Rollback feature.

Within the Migration Status page, records for successful migrations will have an inline “Undo” button. Each “Undo” button is associated to the respective snapshot taken immediately prior to its inline migration. *Records for Rollbacks will also be included within the Migration Status page.*



Rollbacks are permanent actions; it cannot be undone.

All snapshots are permanently deleted upon upgrade of a major Release. Snapshots are not deleted upon upgrade of an Update (dot Release) and thus can still be rolled back to.

Snapshots can fail if data is missing or incorrect.

A snapshot is not saved if a Migration fails, and therefore the “Undo” button is not available.

Migration records cannot be deleted from the Migration Status page if the “Undo” option is still available.

When migrating, Util Libraries that are referenced by selected Commerce Processes and Config Families will be listed in a dependencies dialog pop up. An individual Util Library will be shown even if it is selected in the main window at the start of migration. *It is highly recommended that the user select the Util Library in the dialog pop up even if it has already been selected in the main window.*

Beware of dependencies. **For example:** Migrate Util Libraries used in produce families and processes before, or with, configuration or commerce migration. Upload data tables before migrating Util Libraries, configuration or commerce. Migrate product definition before catalog data. Similarly, migrate configuration data after migrating catalog. For newly created commerce processes, migrate the process before migrating documents. Do not make manual changes to the target site if migration services are available for the category Make the change in the source site and then migrate.

RELATED TOPICS

MIGRATION MANAGEMENT

OVERVIEW

Migration Management has been improved to allow you to easily pull changes from one BigMachines environment to another. To do this, BigMachines features a migration center that allows you to migrate the Util Library, Product Definition, Product Catalog, Attributes and Rules within Configuration, Commerce Process and Documents. In previous versions, test to production was done by downloading XML from the test site and uploading them on to the production site. Much of this Test to Production process has been replaced with Migration. Migration works by pulling data from the source site (test) to the environment you are currently in (production).

Highlights

- One-Touch Migration for commerce processes, documents, product family, catalog, configuration and util libraries
- View differences between the source and target sites prior to migrating
- View detailed differences of code change prior to migrating

---

Migration Basics

When migrating, changes are pulled from the source site to the target site. Commonly, the source site is referred to as 'test' and the target site as 'production'. Since migration works by pulling data fro the source site, the target site is the one that you are logged into and where you're accessing the Migration Center.

Only deployed data can be migrated. Additionally, detailed differences show the differences between the deployed data in both sites. On the target site, migrated components are automatically deployed. For example, if a commerce process is migrated from source to target, the process will be deployed on the target site after the migration is finished.

A component created manually or through bulk services on the source and target instances cannot be considered equivalent migration, even if they have the same variable name. For example, a commerce process on the source site with the variable name "myProcess" is not equivalent to a commerce process created manually on the target site with the same variable name. "myProcess" should have been created on the target site, through migration, for it to be equivalent to the process with the same variable name on the source site. Attempting a migration in this case will throw an error. We advise you to delete the manually created component on the target instance and then perform a migration.

ADMINISTRATION

The Migration Management Center allows you to migrate individual elements of your application in one central location.



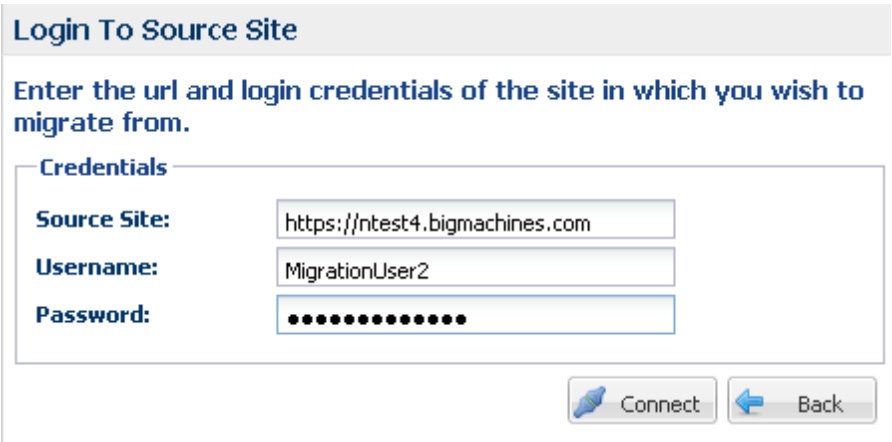
Table of Migration Center Elements

Element	Description
Migrate Button	Click Migrate after you have selected the module(s) that you would like to migrate from a specified source.
Status Button	Click Status to check the migration status. This is also where you will find a link to the summary log that provides more detailed information on successful or failed migrations.
Refresh Button	Click Refresh to re-evaluate the differences between your current environment and the source environment you are migrating from. This will help you determine what modules to migrate. Note: Each refresh calls the source site using system resources to process the data. Clicking refresh many times in a row could slow the loading of the left pane.
Source Site	Displays the URL of the environment you are migrating from.
Util Library	Contains a list of the Util Library functions that can be individually migrated from one environment to another.
Product Definition	Contains a list of the product families that can be individually migrated from one environment to another.
Catalog	Contains a list of the Product Lines and their subsequent Models that can be individually migrated from one environment to another. Note: This only migrates the catalog hierarchy.
Configuration	Contains a list of all the rules and attributes associated with each product family. Included are configurable attributes, rules, configuration and search flows, integration XSL, and stylesheets.
Commerce	Contains a list of all the Commerce Processes that can be individually migrated from one environment to another.
Document Engine	Contains a list of documents, generated by the Document Engine and sorted by Commerce Process, that can be individually migrated from one environment to another.

Logging into the Migration Center

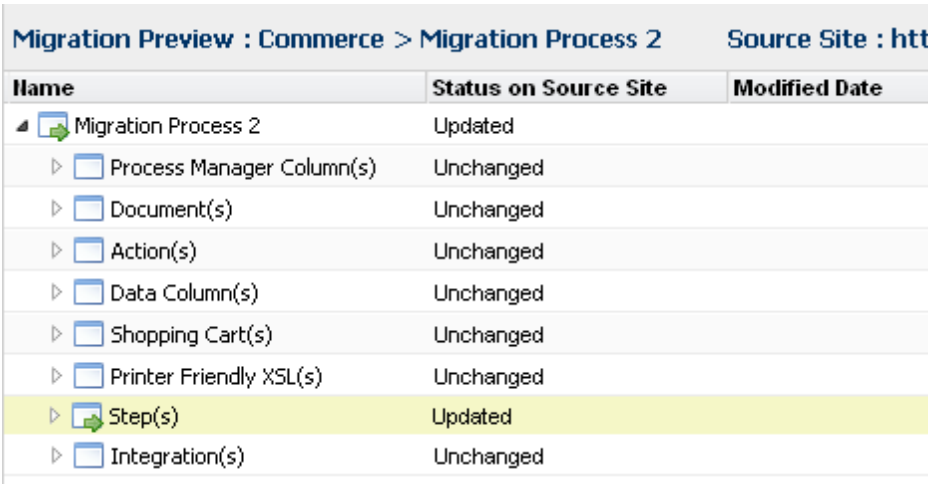
Upon accessing the Migration Center on the target site, you'll be asked to log into the source site. Please note: you can set the source site for a particular BigMachines instance by default. The user must login as a service user or as a host company superuser of the source site. Before you can access the migration management center, you first need to log in to your test environment

1. Navigate to Admin Platform > Utilties > Migration
2. Enter your credentials for the source site (commonly the test site) in the pop-up window.



3. Click **Connect**. A loading box will appear while the connection is being established, refreshing the data from the source site and calculating any differences between the two environment.

4. Upon successful connection, you'll see the Migration Center. In the right panel of the migration repository, you'll see a Migration Preview. If you click on an individual element with a module (or folder), you will see what it contains.



5. In the image above, there are three columns displayed: Name, Status on Source Site and Modified Date. Icons with green arrows indicate that this level may be expanded to display child elements.

Element	Description
Name	These are the names of the elements within the module. You can click the arrow to expand the list to see child elements.
Status on Source Site	This displays the status of each element on the source site when compared to the same element on the current site. 'Updated' incidcates that this element differs between the source site and current site. 'Unchanged' means that the source stie and current site are the same for this element.
Modified	This reflects the date in which the element was modified, inserted or deleted in the source environment.



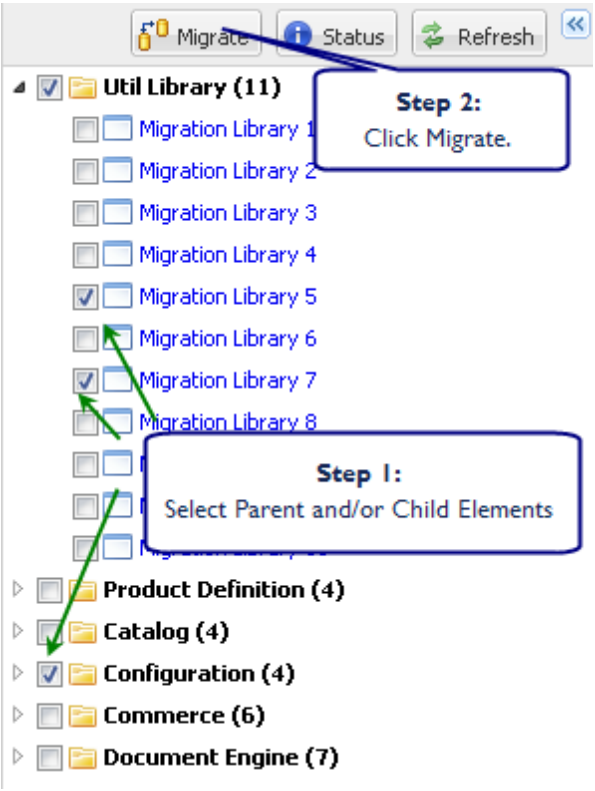
Click on the column header to see filtering options. Each column header allows you to select or deselect the columns you'd like to see. You can also perform an ascending or descending sort Status on Source Site and Modified Date columns

---

Migrating from the Source Environment

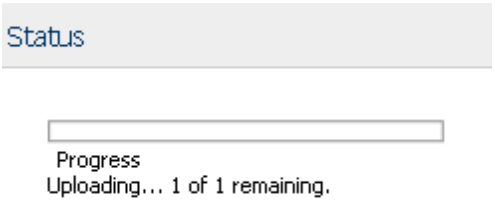
You can select a folder(module) in its entirety, or you can select individual elements you'd like to pull from the source environment.

1. Select Parent and/or Child Elements.



2. Click **Migrate**.

3. A status bar will appear in the bottom of the left panel. For migration of many items, the text provides a real-time indicator of the progress.



A dialog window attached to migration button allows the user to enter a migration description which is then displayed on the Migration Status page for reference.



Source and target sites must be on the same version in order to migrate. When upgrading to a new BigMachines version, make sure that the target and source sites are database copies of one another. First upgrade your test site (source), then upgrade production (target).

Commerce Migration Process

Standard Navigation Path: Administration Home Page > Commerce Processes > Process Definition > Migration

This functionality allows you to map old commerce processes to new commerce processes by selecting a process from each list, and clicking Add. After mapping commerce processes, set up document and workflow step mapping by choosing a mapped commerce process set, and clicking Document Mappings or Step Mappings. After all mappings have been set up, you can migrate commerce process data.



The migration feature is turned on based on a commerce global setting under Administration Home Page > Global System Settings and Utilities > Commerce.

NOTES



Nominate one fullaccessuser to perform migrations.

Remember that items not available in the Migration Center can be migrating by using Bulk Upload/Download. Some examples include: Accounts, Commerce Transactions, Parts and the File Manager.

Establish a process for migration – It is strongly recommended that migrations be performed one-way. For example, always migrate from the test to the production instance and not vice versa. The System fullaccessuser can set it up such that the source site name is auto-populated in the Migration page of the target site.

Entities that are migrated should not be created manually or through bulk upload/download on the target site. Manually created entities, or those added through bulk services, are not considered equivalent for migration, even if they share the same variable name.



Up to 10 snapshots are stored at once. For more information, see Rollback & Snapshots.

Rollbacks will not affect transactions; a Rollback does not delete Steps.

Data Columns can be deleted through Migration. *Deletion of a Data Column via Migration will have the same effect on Transaction Manager Views and Reports on the target site as if it were deleted in a Rollback.*

Only users with access to perform Migrations can also perform Rollbacks.

Status emails are sent to the Superuser after Rollback is complete.

Users will experience an increase in total Migration time due to the addition of the snapshot that must be taken to facilitate the Rollback feature.

Within the Migration Status page, records for successful migrations will have an inline “Undo” button. Each “Undo” button is associated to the respective snapshot taken immediately prior to its inline migration. *Records for Rollbacks will also be included within the Migration Status page.*



Rollbacks are permanent actions; it cannot be undone.

All snapshots are permanently deleted upon upgrade of a major Release. Snapshots are not deleted upon upgrade of an Update (dot Release) and thus can still be rolled back to.

Snapshots can fail if data is missing or incorrect.

A snapshot is not saved if a Migration fails, and therefore the “Undo” button is not available.

Migration records cannot be deleted from the Migration Status page if the “Undo” option is still available.

When migrating, Util Libraries that are referenced by selected Commerce Processes and Config Families will be listed in a dependencies dialog pop up. An individual Util Library will be shown even if it is selected in the main window at the start of migration. *It is highly recommended that the user select the Util Library in the dialog pop up even if it has already been selected in the main window.*

Beware of dependencies. **For example:** Migrate Util Libraries used in produce families and processes before, or with, configuration or commerce migration. Upload data tables before migrating Util Libraries, configuration or commerce. Migrate product definition before catalog data. Similarly, migrate configuration data after migrating catalog. For newly created commerce processes, migrate the process before migrating documents. Do not make manual changes to the target site if migration services are available for the category Make the change in the source site and then migrate.

RELATED TOPICS

DOCUMENT MIGRATION

OVERVIEW

Two document migration methods:

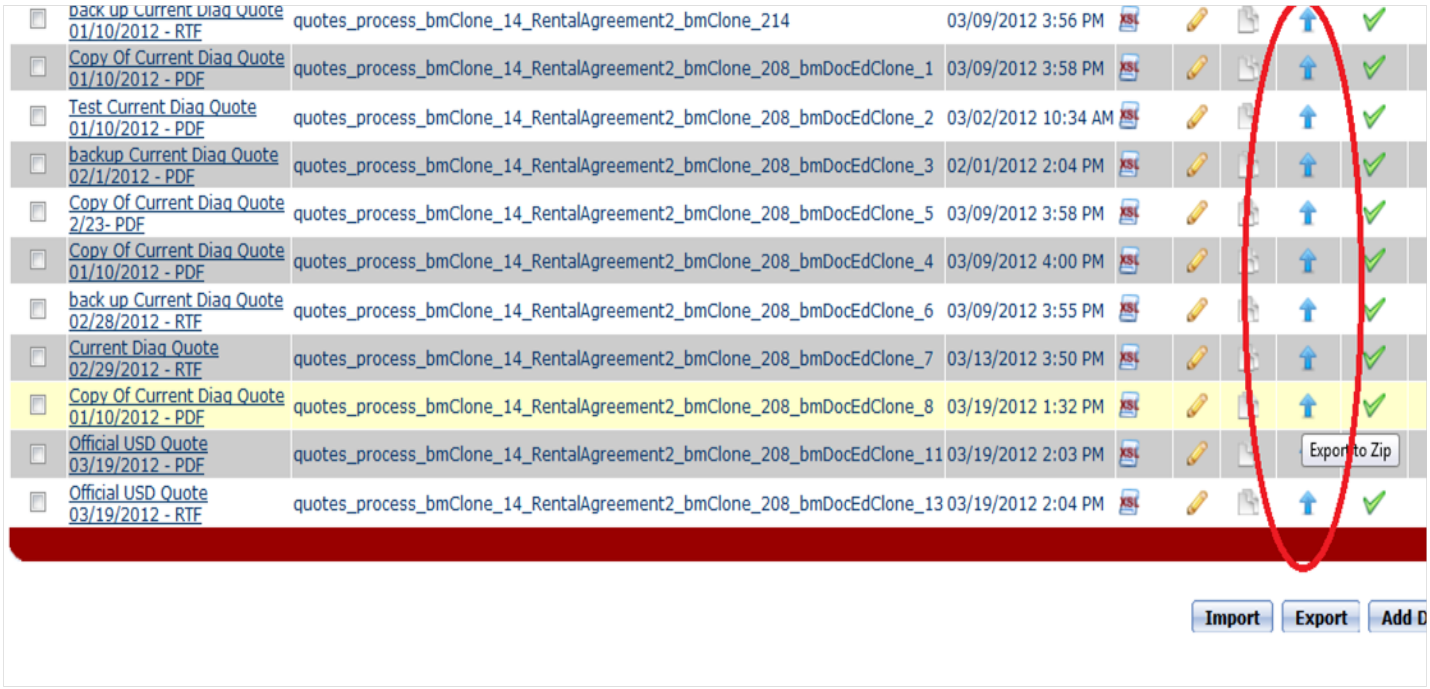
1. "Updating" the document, which is where your source and target instances have been refreshed with each other, and are in Sync. For "Updating", it only works if you edited an existing document in your source instance, and when you export and migrate it to your target instance, there must be an old copy of that document in there (from the refresh) in Admin > Documents. Otherwise BM will give you an error that source cannot be found. This is the default method BM will use for migrating documents.

2. If your document does not exist on the target site, you would need to "Add" it. We would have to export your file, and edit one of the files to tell the system that we want it to "Add" the document instead of "Update" it. **Note:** Your source and target instances must still have been refreshed with each other, otherwise your commerce system will be out of sync and this will not work either.

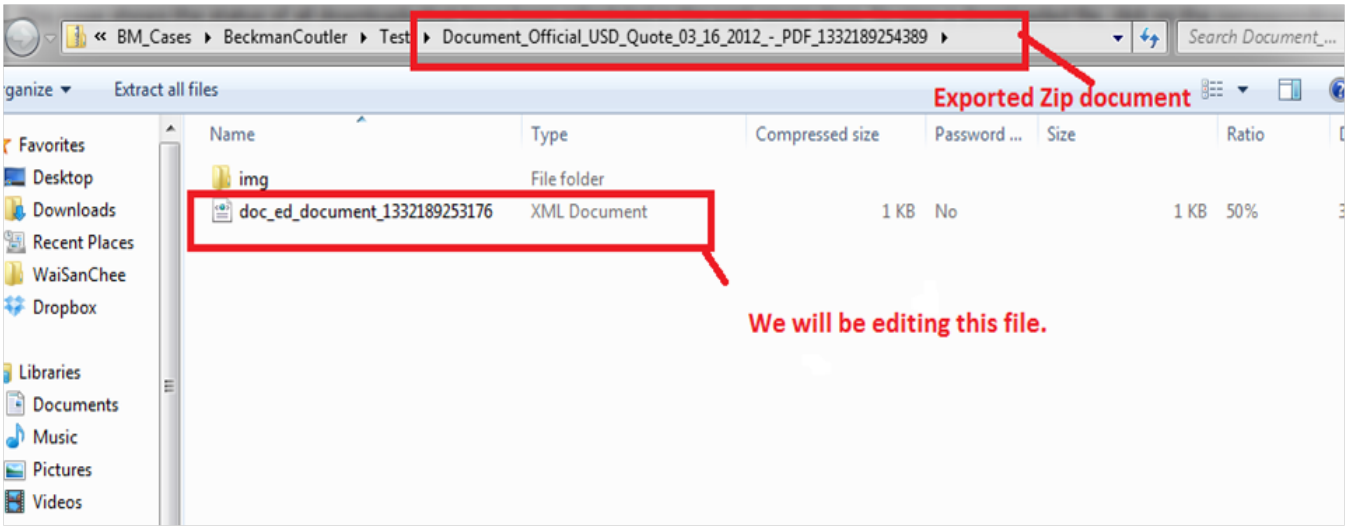
ADMINISTRATION

Migrating a Document

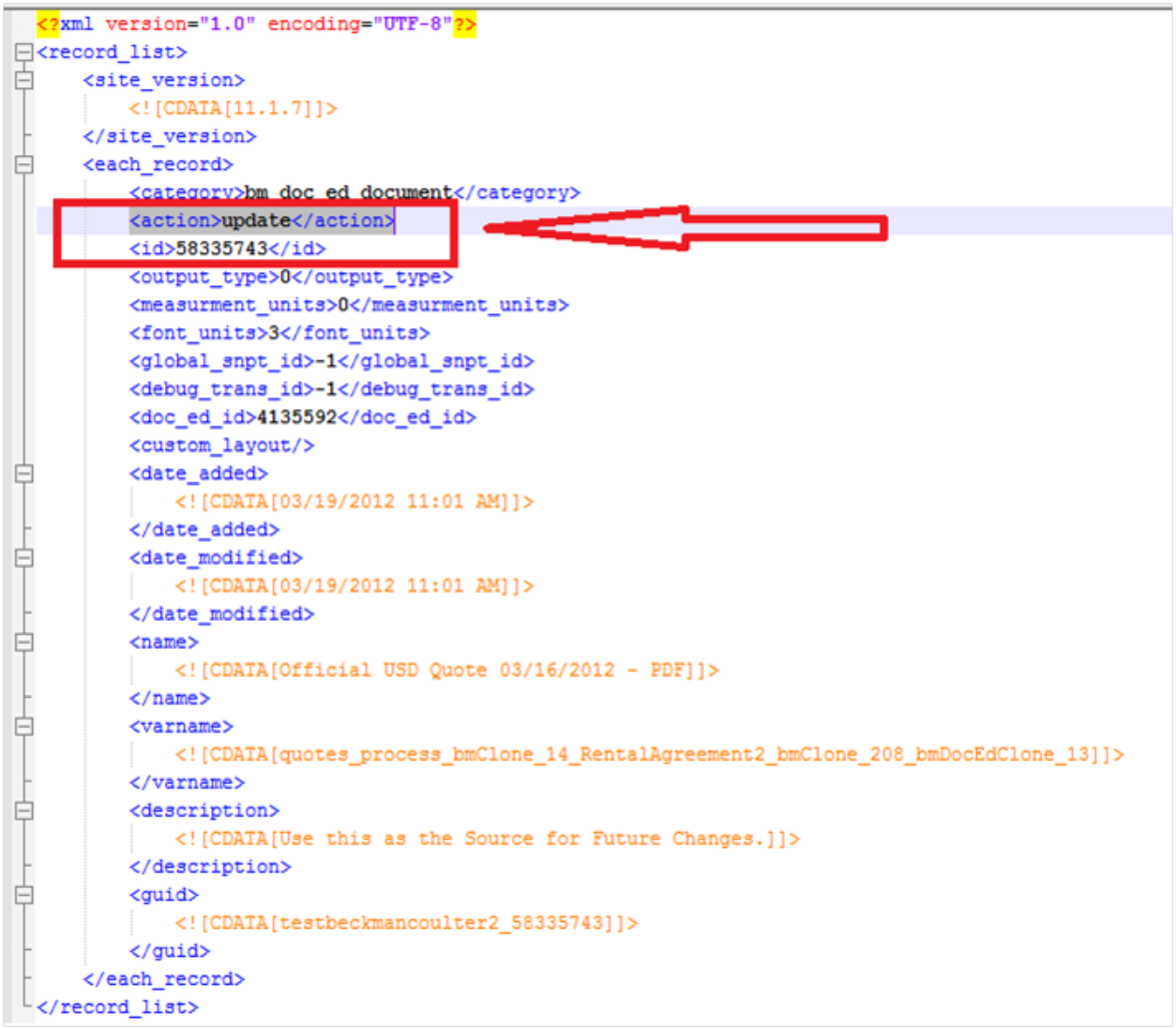
1. Export the document.



2. Once you've saved it from the download page, open up the zip file and you should see:



3. The second xsl file is the one we need to edit. Open this up in any code editor you use, and search for <action>update</action>:



4. Change it to: <action>add</action>
5. We also want to blank out the <id></id> field. This ID field contains the transaction ID of the quote that is from your source site, so it might throw errors when trying to import this information.
6. Save this file, and re-zip it with your newly edited for and the “img” folder (this contains the actual xsl code for your document) and Import it into your target site.

NOTES



A new dialog window attached to migration button allows the user to enter a migration description which is then displayed on the Migration Status page for reference.

Data Columns can be deleted through Migration.



When an Admin initializes a migration, the system automatically stores a “snapshot” of the destination site prior to performing the migration. Snapshots allow you to Rollback to a previous, more stable state. For more information, see Rollback & Snapshots.

Users will experience an increase in total Migration time due to the addition of the snapshot that must be taken to facilitate the Rollback feature.

Only users with access to perform Migrations can also perform Rollbacks.

Status emails are sent to the Superuser after Rollback is complete.



All snapshots are permanently deleted upon upgrade of a major Release. Snapshots are not deleted upon upgrade of an Update (dot Release) and thus can still be rolled back to.

Snapshots can fail if data is missing or incorrect.

A snapshot is not saved if a Migration fails, and therefore the “Undo” button is not available.

Migration records cannot be deleted from the Migration Status page if the “Undo” option is still available.

When migrating, Util Libraries that are referenced by selected Commerce Processes and Config Families will be listed in a dependencies dialog pop up. An individual Util Library will be shown even if it is selected in the main window at the start of migration. *It is highly recommended that the user select the Util Library in the dialog pop up even if it has already been selected in the main window.*

RELATED TOPICS

WHAT CAN BE MIGRATED?

OVERVIEW

What can be migrated?

The table below provides a list of the categories and components that can be migrated. It will also tell you if you can view detailed differences.

Element	Description	Detailed Differences	Notes
Util Library	Contains a list of the Util Library functions that can be individually migrated from one environment to another.	Available	• Add and modify selected util libraries through migration. • Delete is not permitted through migration.
Product Definition	Contains a list of the product families that can be individually migrated from one environment to another.	Not Available	• Add and modify specific product families through migration. • Regular attributes can be deleted through migration.
Catalog	Contains a list of the Product Lines and their subsequent Models that can be individually migrated from one environment to another. Note: This only migrates the catalog hierarchy.	Not Available	• Add and modify product lines and models in specific product families. • Product lines and models can be deleted through migration, but Product Families cannot be deleted through migration.
Configuration (See Notes)	Contains a list of all the rules and attributes associated with each product family. Included are configurable attributes, rules, configuration and search flows, integration XSL, and stylesheets. All Product Family level attributes, rules and stylesheets are included.	1. View detailed differences between rules by clicking "Configuration". 2. View detailed differences of code changes.	• Add and modify attributes, rules and layout in specified product families. • Attributes and Rules can be deleted through migration, but Product Families cannot be deleted through migration.
Commerce (See Notes)	Contains a list of all the Commerce Processes that can be individually migrated from one environment to another.	1. View detailed differences between actions, attributes and rules 2. View detailed differences of code changes.	• Add and modify all component elements of a specific commerce process through migration. • Components can be deleted via migration, but commerce processes cannot be deleted.
Document Engine	Contains a list of documents, generated by the Document Engine and sorted by Commerce Process, that can be individually migrated from one environment to another.	Not Available	• Add and modify documents through migration. • Documents cannot be deleted via migration.
Data Tables (See Notes)	Contains a list of all data tables that can be individually migrated from one environment to another.	1. View low-level differences between data	• Last modified date and modified by (user who most recently made changes to the data table prior to

Element	Description	Detailed Differences	Notes
		tables. 2. View data table scheme changes	deploying. Please note that data changes will not be shown.
File Manager	Contains a list of files and files folders that can be migrated one at a time.	Not Available	

What can't be migrated?

The following cannot be migrated using the Migration Center.

Component	Sub-Component
Accounts	
Parts	• Part Custom Fields & Filters • Part Associations • Pricebooks • Favorites
Serial Number area	*Includes custom fields*
Users & User Groups	
Partner Organizations	
Header & Footer	
Stylesheets	
Page Templates	
Text Administration	
Home Page	• Punch-Ins • Access Rights
Mobile Home Page	• Images
Process Invocations	
Native Reports & Commerce Manager Views	
Integration Settings	

Settings	• Config Settings
	• Commerce Process Settings
	• General Settings, including:
	  ○ Countries & States
	  ○ Email Signature
	  ○ Favorites
	  ○ General Site Options (Navigation Menus)
	  ○ Host Company

Transactions	

NOTES

Please see relevant notes related to the migration of components or categories.

Product Definition/Catalog/Configuration

- All deployed changes within the selected product family are migrated. Changes to product definition, or catalog, must be made prior to migrated configuration data.
- You must manually add support for a newly added product family in Catalog Definition before migrating catalog or configuration for that product family. Please read this article and follow steps 10-15.
- Please contact your System fullaccessuser to make a product family deployable.

Commerce

- Clicking the name of modified commerce processes allows you to view the specific components of the commerce process. If only a translation has been added or modified, the change is not shown at this level. Translations will be migrated when the commerce process is migrated.
- If the target site has a deployed step, commerce document or data column that is not present or un-deployed on the source site, migration will fail. This is to ensure that the end user data remains consistent after a migration.

Data Tables

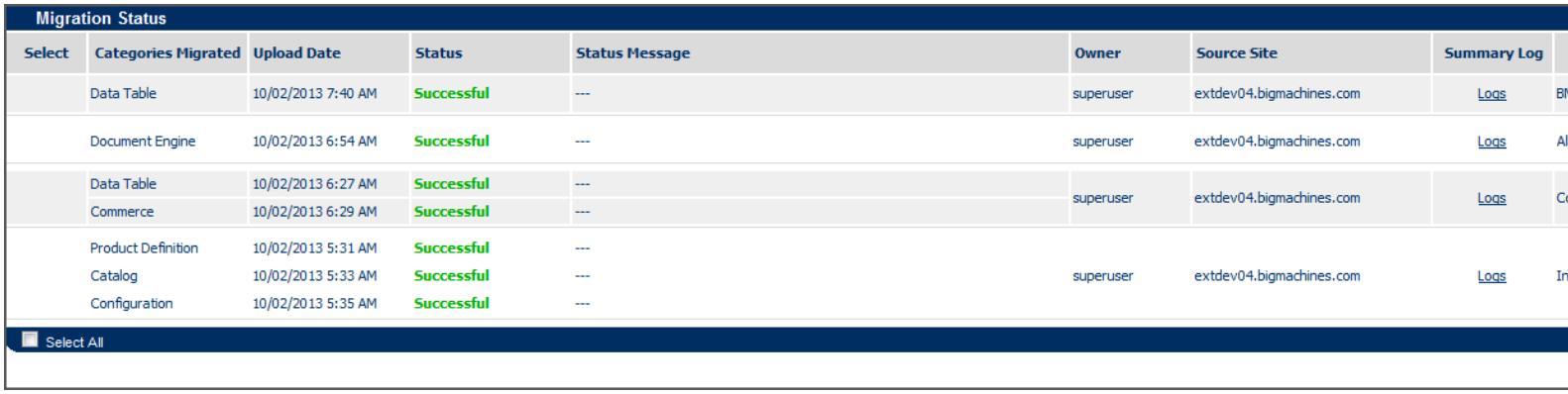
- When migrating configuration or commerce, a list of dependent data tables will be displayed in a pop-up window. If these data tables are not selected, migration will fail. This pop-up will only appera when menu attribute validations are mapped to a data table, but not when the data table is used in rules using BMQL.
- When migration changes attribute menu values, if you delete an attribute or menu value, the link will be removed from the data table.

RELATED TOPICS

ROLLBACK & SNAPSHOTS

OVERVIEW

Rollback allows you to undo changes on a target site that were previously migrated; it returns the target site to a stable state. If something unwanted is migrated or you experience unexpected behavior, the target site can be Rolled back to a previous state, known as a "snapshot". *Like standard migrations, Rollbacks will not affect transactions.*



Snapshots

When an administrator initializes a migration, the system automatically stores a “snapshot” of the destination site prior to performing the migration. Snapshots include all migrate-able categories, with the exception of Data Tables and File Manager files that were not selected for that respective migration. *Snapshots capture only the deployed version of each category.*

Within the Migration Status page, records for successful migrations will have an inline “Undo” button. Each “Undo” button is associated to the respective snapshot taken immediately prior to its inline migration. Upon selecting an “Undo” button, the target site will revert back to the state captured by the associated snapshot, i.e. the state immediately prior to that migration.



Records for Rollbacks will also be included within the Migration Status page.



Migration records with the “Undo” action available cannot be deleted.

Versioning

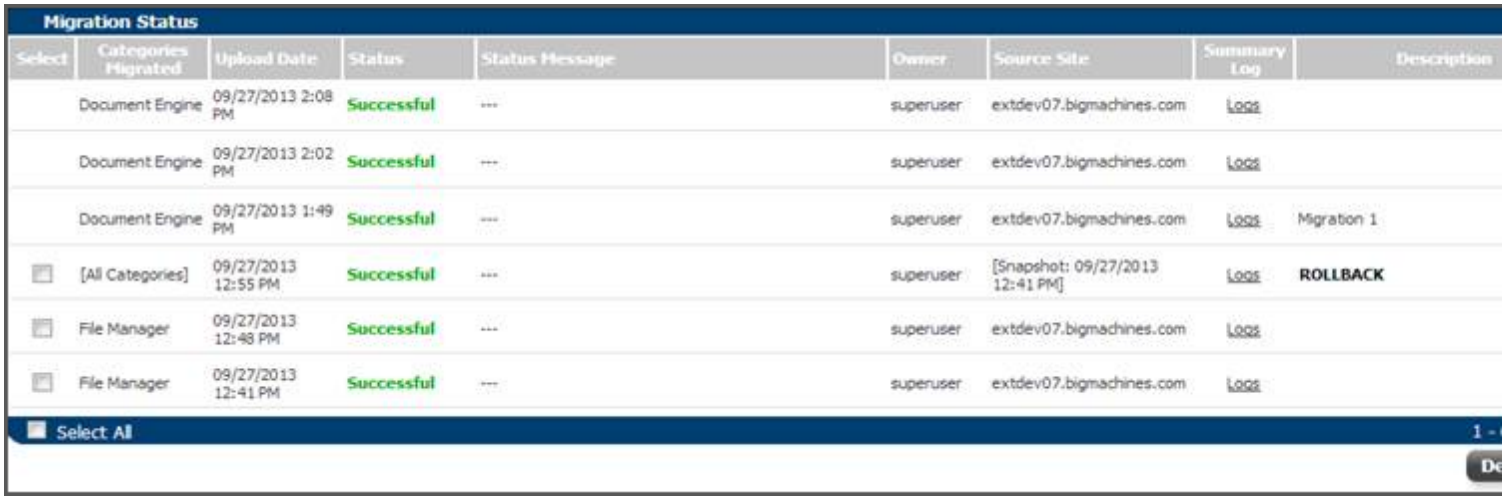
A backup of the previous version of a product family is taken every time its deployed. This functionality does not back-up associated data like data tables or Util Libraries.

The previously deployed version of a configuration product family is backed up every time a deployment occurs. The first deployment after a product family is created does not create a backup because no previously deployed versions are available. Since configuration migration performs an automatic deployment, a backup of the segment always occurs after a successful migration. *One version (last deployment) of the product family is maintained at a time.*

ADMINISTRATION

Rolling back to a Snapshot

1. Log in to the Migration Center. For more information, see Migration Management.
2. Locate the snapshot of the previous state that you would like to Rollback to.



3. Click **Undo**.



4. Once Rollback is completed, click **Refresh** or **Back**.



Rollback is a permanent action. Rollbacks cannot be undone.



While a Rollback is in progress, users cannot trigger: Migration, Deployment, or additional Rollbacks.

Rollback & Steps

Rollbacks do not delete Steps. If a Step exists on the target site in current state but does not exist in the snapshot selected to roll back to, the Step will remain on the target site with the following effects:

- All Transactions in the Step will remain untouched by the system.
- Transition Rules and Forwarding Rules under the Step will remain untouched. Note that this will allow any transaction in the Step to still transition out as it would have at the time Rollback was initiated.
- Transition Rules and Forwarding Rules for all other Steps will be reverted to the state of the snapshot. **Note:** because the Step did not exist at the state of the snapshot and thus no rule could have existed to allow a transition into the Step, transactions will no longer transition into the Step after Rollback without direct Admin changes or an additional migration of Commerce.
- All Participant Profiles, User Access Rights, and Document Views for the Step will remain untouched by the system.

Rollback & Data Tables/File Manager

Individual Data Tables or File Manager files are affected in Rollback only if a version is contained within any snapshot between current state and the state being rolled back to.

- An individual Data Table will only be included in the snapshot taken immediately prior to a migration if it has been selected for that specific migration. Therefore, this Data Table may be contained within multiple snapshots, only one snapshot, or no snapshot at all. *The same follows for an individual File Manager file.*
    - A Data Table will be rolled back if it is contained within a snapshot (i.e. has been migrated) between present state and the selected snapshot. If a Data Table is contained within multiple snapshots, it will be rolled back to the version stored within the oldest snapshot in this timespan.
    - A File Manager file will be rolled back if it is contained within a snapshot (i.e. has been migrated) between present state and the selected Rollsnapshot. If a File Manager file is contained within multiple snapshots, it will be rolled back to the version stored within the oldest snapshot in this timespan.



Rollbacks will not fail if a simple table-based rule is reverted when using a table that no longer exists. *A warning message will be logged in the Rollback status.* Simple table-based rules will return null at run time. When the administrator next attempts to edit the rule, the application will request that the user select a new table.



Any Data Table or File Manager file not contained within a snapshot between current state and the snapshot being rolled back to will not be deleted nor altered by a Rollback.

Rollback & Data Columns

Rollback will remove a Data Column(s) if it is not contained within the snapshot selected for Rollback.

- If a Data Column is removed via Rollback and was referenced in the Process Manager at the time Rollback is initiated, it will be removed from the Transaction Manager columns.
- If a Data Column is removed via Rollback and was referenced in the Process Manager at the time Rollback is initiated, it will be removed from any custom Transaction Manager Views.
    - Affected views will be highlighted red in the drop-down menu.
- If a Data Column is removed via Rollback and was referenced in a Report prior to Rollback, the Report will become invalid.
    - The invalid Report name will be highlighted red in the Report Manager.
    - The creator must manually remove deleted Data Columns used in the Report before the Report can be saved, scheduled, or run.
        - "[Attribute removed]", highlighted in red, appears where a reference to a deleted Data Column was used.
    - The system will send an email to the creator of any report that was made invalid by removed Data Columns during the Rollback process.
        - Scheduled Report emails will notify the recipient that errors have occurred and they must contact their admin.



Data Columns can be deleted through Migration. *Deletion of a Data Column via Migration will have the same effect on Transaction Manager Views and Reports on the target site as if it were deleted in a Rollback.*

Snapshot Storage

The system will store up to 10 snapshots. Accordingly, the Migration Status page will contain up to 10 migration records with an inline "Undo" button.

- Snapshots are stored for successful migrations only. The snapshot taken immediately prior to a migration will not be stored if the migration is unsuccessful.

- If the number of snapshots attempting to be stored reaches 11 and, accordingly, there would be 11 migration records with an "Undo" action present, the oldest snapshot will be permanently deleted from the database and the 10 most recent snapshots will remain. This deletion will occur upon completion of the next successful migration.
- Once a snapshot is selected for Rollback, i.e. "Undo" has been selected for the inline migration record, that snapshot and all other snapshots that were taken after the selected snapshot will be permanently deleted.

NOTES



All snapshots are permanently deleted upon upgrade of a major Release. Snapshots are not deleted upon upgrade of an Update and thus can still be Rolled back to.

Because Rollback performs similarly to Migration, when dependencies are removed for the migration and then a Rollback occurs, the results are equivalent to migrating without those dependencies.

Rollbacks will not fail if a simple table-based rule is reverted when using a table that no longer exists. *A warning message will be logged in the Rollback status.* Simple table-based rules will return null at run time. When the administrator next attempts to edit the rule, the application will request that the user select a new table.



An increase in total Migration time will occur due to the addition of the snapshot, which must be taken to facilitate the Rollback feature.

Only users with access to perform Migrations (e.g. Superuser and FullAccess user with Superuser permissions) can also perform Rollbacks.

Status emails are sent to the Superuser after Rollback is complete.

Migration logs include errors produced when taking a snapshot.



Rollback is a permanent action. Rollbacks cannot be undone.

While a Rollback is in progress, users cannot trigger: Migration, Deployment, or additional Rollbacks.

A snapshot is not saved if a Migration fails, and therefore the "Undo" button is not available.

Migration records cannot be deleted from the Migration Status page if the "Undo" option is still available.



Snapshots can fail if data is missing or incorrect. A Rollback will fail if a Commerce or Config attribute that is used in a data table validation is reverted. *An error message will state that the attribute does not exist. User must manually remove the validation from the target site.*

When migrating, Util Libraries that are referenced by selected Commerce Processes and Config Families will be listed in a dependencies dialog pop up. An individual Util Library will be shown even if it is selected in the main window at the start of migration. *It is highly recommended that the user select the Util Library in the dialog pop up even if it has already been selected in the main window.*

RELATED TOPICS

SHOWING DIFFERENCES

OVERVIEW

You can click the Refresh button to show the difference between the source site and the destination site for some of the modules and their respective elements. You can also perform this function on individual module elements.

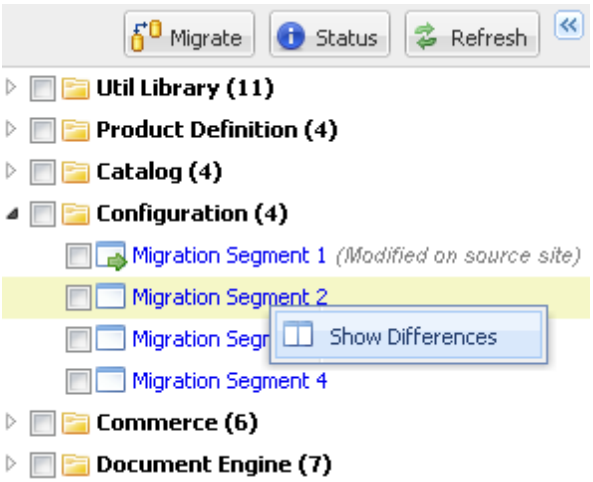
ADMINISTRATION

Showing differences at the element level

1. Select a **Folder (Module)** to expand.

> If the element is BLUE, that means that you can check for differences. If an elements is shown in BLACK, you will be unable to see changes.

2. **Right-Click** on the element that you'd like to check for differences.



3. Click **Show Differences** (or double-click the module name). A loading dialog box will appear while the application is checking the source site. The element that you selected will display in the right-panel under Migration Preview

You may also notice a green arrow or a red minus icon next to certain elements. If this occurs in the RIGHT panel, the green arrow means that an item exists on the source site that doesn't exist on the production (or current) environment. The red minus means an item doesn't exist on the source site that does exist in destination (or current) environment.

If this occurs in the LEFT panel, the results are similar to how it works in the right panel, however, if it is insertion or deletion, you will see the final results of the item once the migration has been completed.

NOTES



For functions (Util Library, Modify Action Functions, etc.), you can perform a low-level comparison which will show you a side-by-side review of the lines of code that are different between the two environments.

RELATED TOPICS

BUILD A REPORT

OVERVIEW

This article assumes you have already created a new report or you are editing a report.

Each Report can have two types of criteria: Standard and Advanced. Results are generated based on transactions that satisfy both of these filter requirements.

The Display Columns section allows the user to build the data that is displayed when the report is run. The user can select attributes to display and specify the way in which the columns must be grouped. The user can choose to display some values that are calculated from numeric/currency attributes that are returned.

ADMINISTRATION

Standard Filters



These filters include the following:

Field	Description
Currency	You can filter by document currency. Reports can return transactions in ALL currencies, as long as the user does not perform any operation, like sum, average, etc. on a Currency type attribute, use that attribute in a calculated column, or use a Currency type attribute in a Group By. In these cases, the user must select a specific currency.
Date Filter	This filter is mandatory. Standard dates that can be used to build reports are Date Created and Date Last Modified. Additionally, users can choose any custom commerce Date attributes that have been mapped to the Data Columns, i.e. Expiration Date.
Duration	This filter is mandatory. You can choose the duration for the date chosen in the Date Filter.

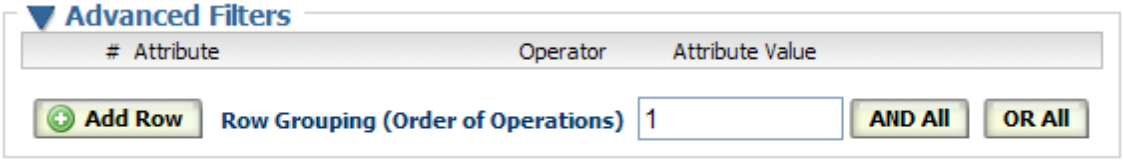


In relation to duration, remember that a week is considered to begin on a Sunday and end on a Saturday. You are allowed to set the beginning of a fiscal years in the Process Administration editor. All Fiscal durations will be based on this date.

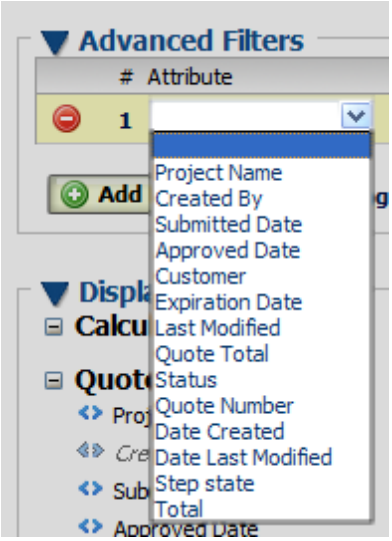
Creating Advanced Filters

The user can define advanced filters on all main-document, sub-document and configurable Text, Numeric, Currency, Date and Menu type attributes that have been mapped to Data Columns.

1. Click **Add Row**.

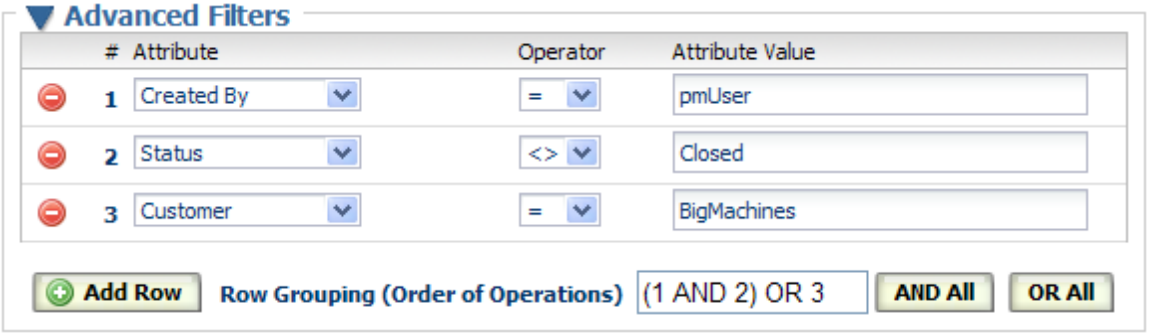


2. Click on the drop-down menu under Attribute and choose the attribute you'd like to filter by.



3. Select the **Operator** and enter an **Attribute Value**.

4. Set the **Row Grouping**, for multiple attributes. If left blank it will default to AND ALL.



Blank values are supported in Filters just as they are supported in saved searches. However, the user cannot create a report without specifying an operator for a filter, unlike in saved searches.

Setting Displayed Columns

Once you have created your standard and advanced filters, you will need to define at least one main-document display column in order to run the report. Failure to do so will result in an error message.

You can select from three options:

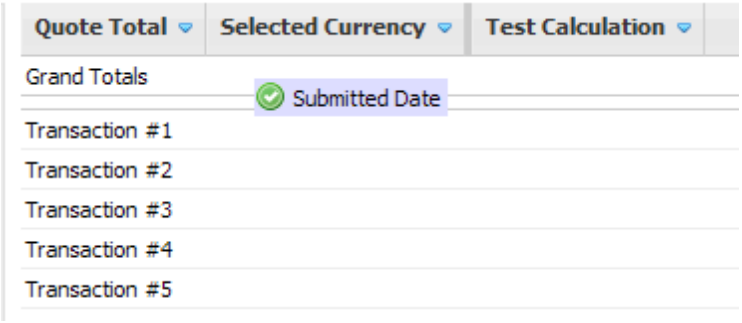
- **Calculated Columns:** This opens a BML-Like editor that allows you to write a script to define the output of the calculated columns.
- **Quote Attributes:** These are the same attributes available in Advanced Filters. All text, numeric, currency and date commerce attributes that are mapped to Data Columns.
- **Line Item Attributes:** These are the line(sub-doc) attributes that are mapped to Data Columns.

---

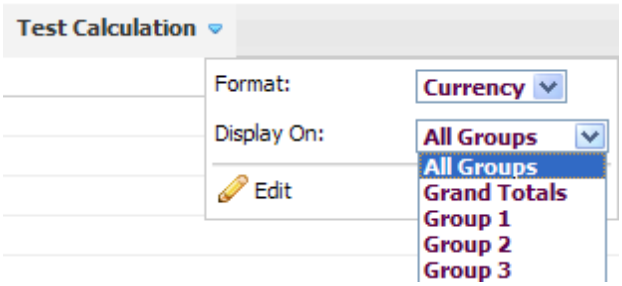
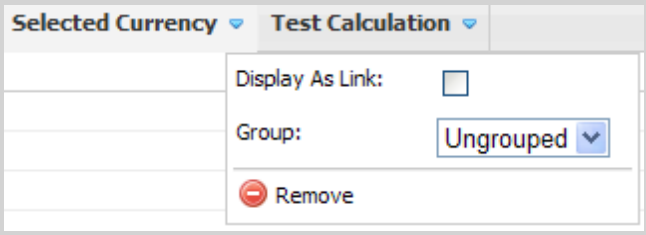
1. Select an attribute to drag-and-drop into the Display Columns panel.
2. A green arrow next to the attribute name means that the attribute can be placed in the position you've selected. A red NO sign will appear next to the attribute if it is not allowed.

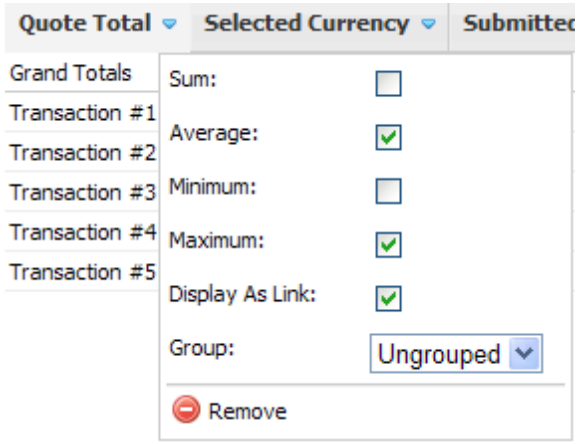


The order you place your columns is how they will appear on the report.

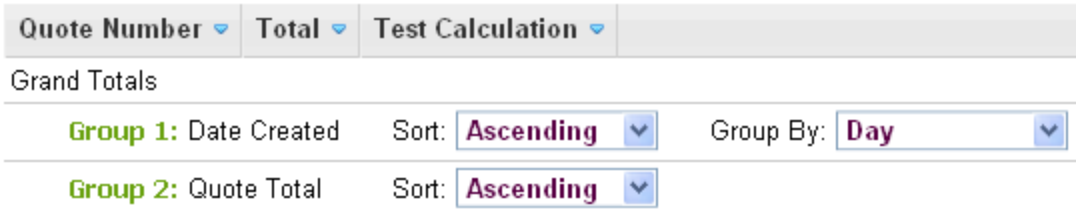


3. Set display column options:

Column Type	What Does It Look Like?	Options	Notes
Calculated Column		• **Format:** Switch between the Number and Currency formats. If the Number format is selected, another option will appear allowing you to set the number of decimal points. • **Display On:** Specify at which group level the column should be displayed. • **Edit:** Make changes to the calculated column.	
Non-Numeric Quote and Line Item Attributes		• **Display as Link**: Displays the attribute as a link that when clicked, takes you to the quote. • **Group:** Determine if this attribute should be used to group the rows. Doing so will remove the attribute as a Display Column. For example, you can group quotes by their Submitted Date or by Selected Currency. • **Remove**: Click to remove the column from the Display Columns panel.	

Column Type	What Does It Look Like?	Options	Notes
Numeric Quote and Line Item Attributes		• **Sum:** Reflect the sum of the aggregate for an attribute. I.e., Quote Totals. • **Average:** Reflect the average for an attribute. • **Minimum:** Reflects the minimum value of all Quote Totals for a group. If there are three quotes ($145,694.23, $34.56, and $35, 434), the minimum will find $34.56. • **Maximum:** Reflects the maximum value of all Quote Totals for a group. If there are three quotes ($145,694.23, $34.56, and $35, 434), the maximum will find $145,694.23.	All numeric quote and line item attributes have the ability to be displayed as aggregates. Aggregates are used in conjunction with Groups to provide data on the x-axis of charts. in this example, you will see the options for the attribute Quote Total.

4. **Group your attributes into rows**. This will remove them from the Display Columns header and place them below.



5. You will then be given the options to sort in Ascending or Descending orders. Depending on the attribute, you may also see options to Group By certain specifics. If you notice the example below, there is a Group By Drop down menu for Group 1 which contains options such as grouping by day, week, month, quarter, year and more.

NOTES



Grouping becomes very important if you plan on adding charts to your report.



Pagination of reports occurs at the group level.

Results within a group will be sorted in ascending or descending order for text, menu, currency and numerical attributes. Additionally, by Duration for date attributes.



All commerce attributes used in reporting must be mapped to Data Columns. HTML, RTE, File Attachment, composite attributes, history fields and approval-related attributes are not available for reporting.

RELATED TOPICS

REPORTING MANAGEMENT

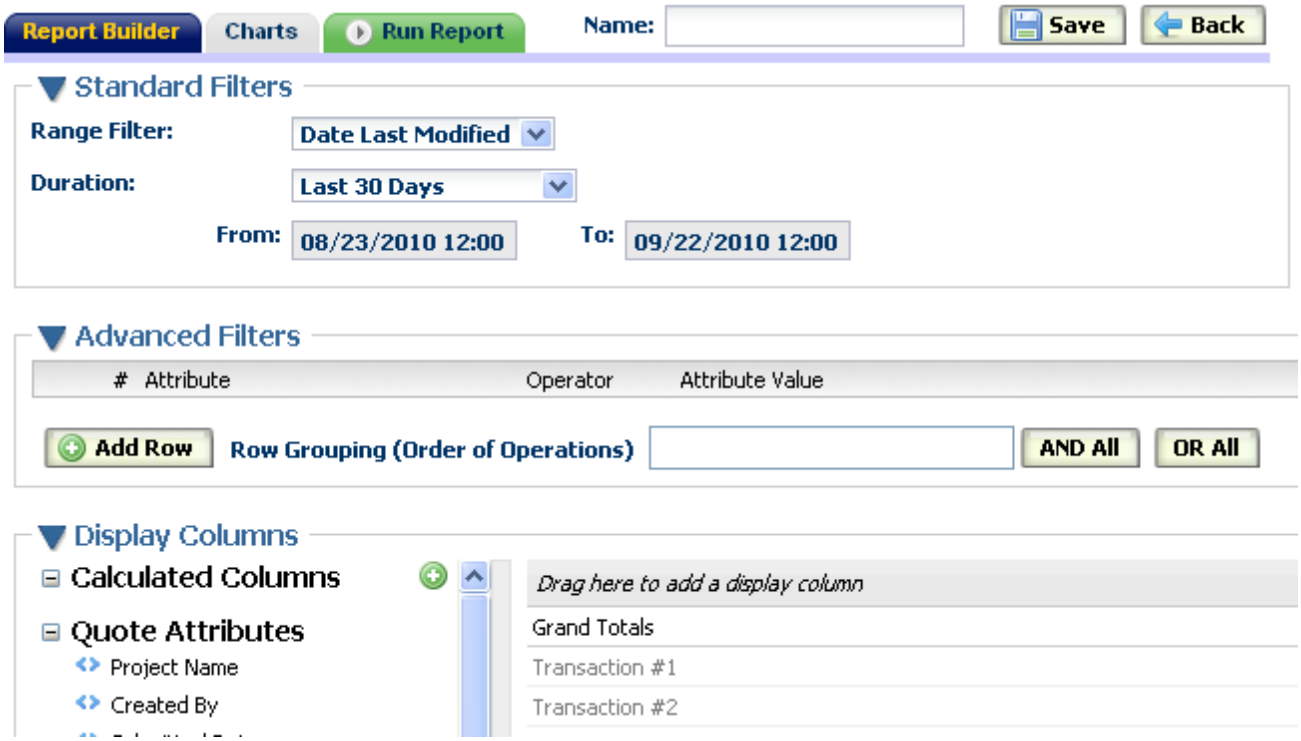
OVERVIEW

BigMachines allows you to generate and store reports. You can generate reports on commerce main document and line item attribute data. Reporting will return results based on user permissions.

A Reporting Manager will be made available for every commerce process in the system. Users can store, edit, run and manage reports in the Reporting Manager. Folders are made available for organizing reports within the Reporting Manager. All users can choose to Share their reports with other users. The other users can run the report on the transaction data that is available to them depending on their user rights. Reports can be built from scratch or from saved views. When a report is run, the results generated are displayed on the UI. These results can also be downloaded to Microsoft Excel. The results can also be sent as an Excel attachment to an email.

Example

You can specify filter attributes to generate report results. For example, you can create and run a report that will give you all the "Closed" quotes that have been created in the last 60 days. In this example, you specify the Quote Status, the Date Created and the Duration (Last 60 days) as the filters to generate the report.



Report Manager Overview

There is one Report Manager available for each commerce process. Using this page, you can add, delete, edit, run, share, schedule and/or store reports.

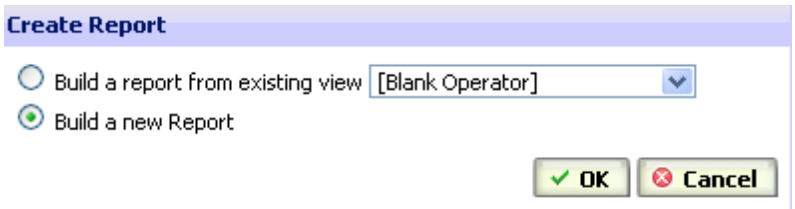


Field or Action	Description
Delete	Click on the delete icon next to the report(s) you wish to delete. Only self-created reports can be deleted. Shared reports must be deleted by the user that created it.
Edit	Click on the edit icon next to the report you'd like to edit.
Report Name	Click on the Report Name Label to edit.
Description	Write a description of the report by clicking on the description field.
Last Modified	Time stamp for the last time the report was updated or edited and saved.
Shared	Check this box to make sure your report is availble to all other users.
Run	Click the run button to generate the report.
Schedule	Click the schedule icon to open the Schedule Report Editor and determine how often to run the reports and to whom the system should email it.
Add	Click the Add button to create a new report.
Update	Click Update after making changes to the description or checking/un-checking the Shared option.
Stylesheet	Click the Stylesheet to open the Reporting Stylesheet Editor.
Refresh Reporting Data	Click Refresh Reporting Data to refresh the reports with the most up-to-date information.

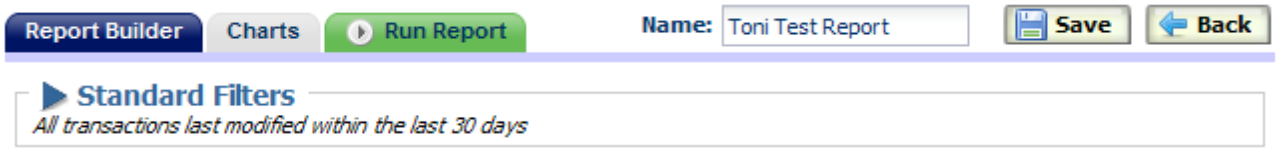
ADMINISTRATION

Adding a Report

1. Click **Add** from the Report Manager to open the Create Report pop-up.
2. Choose an option:
   - **Build a report from existing view:** Choose a report from the drop-down menu.
   - **Build a new report**

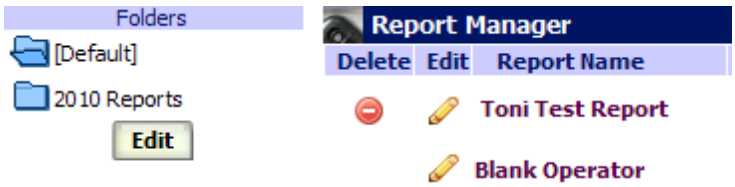


3. Click **OK** to open the Report Builder.
4. Enter the **name of the report** and begin customizing your report.

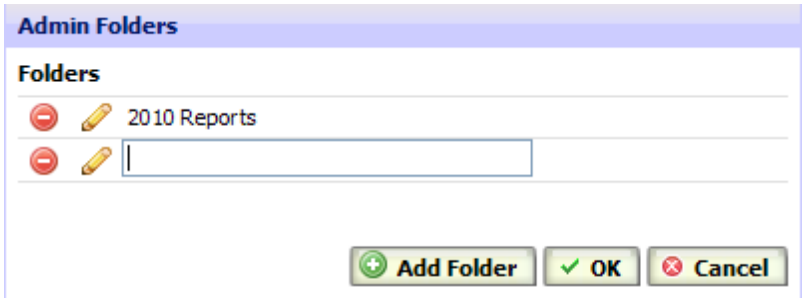


Adding a Folder

1. Click **Edit** under the Folders section within the Report Manager.



2. Click **Add Folder** from the Admin Folders pop-up.
3. Enter the **Folder Name** in the text box.

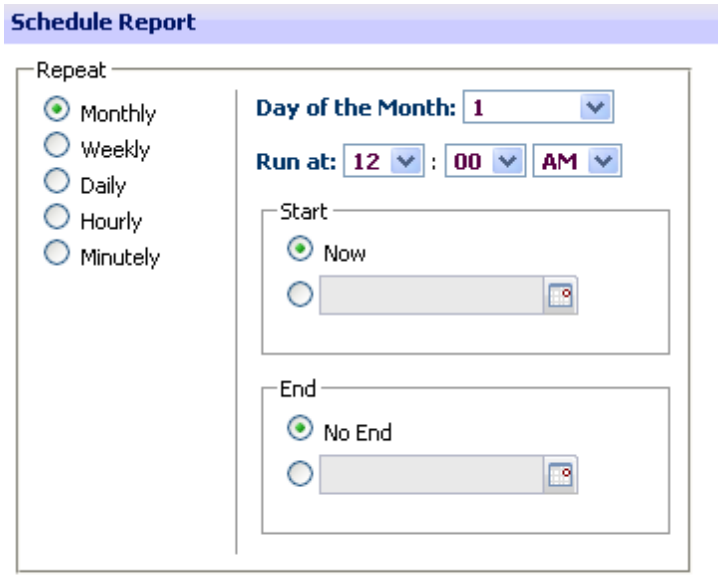


4. Click **OK** to save the change or Cancel to return to the Report Manager.

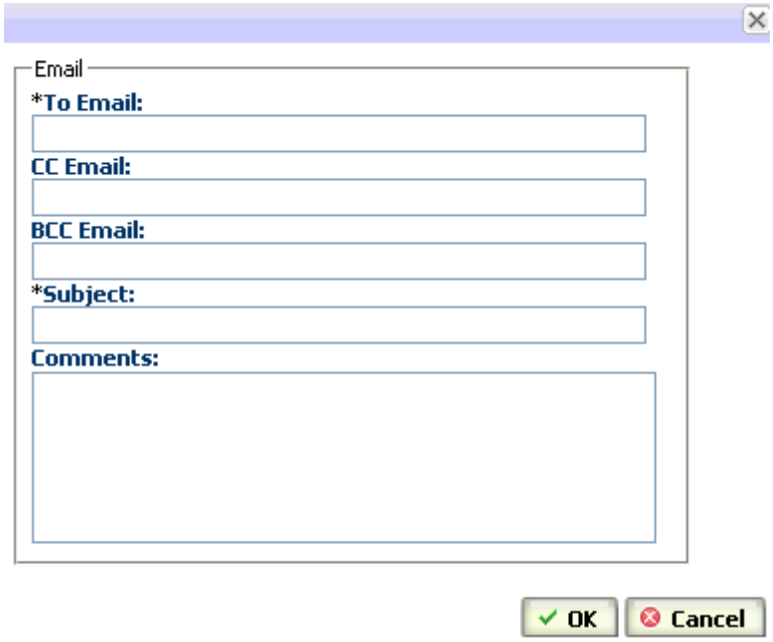
> To edit a folder, click the Pencil icon. To delete a folder, click the Delete icon.

Scheduling a Report

1. Click on the **Schedule icon** on the Report Manager page.
2. Select an option, under Repeat, based on how often you'd like the report to run.
3. Choose the day, week, month, and/or specific time or date for when you will run the report.



4. Enter the email credentials and a subject line for who should receive an email and a report attachment once the report is run.

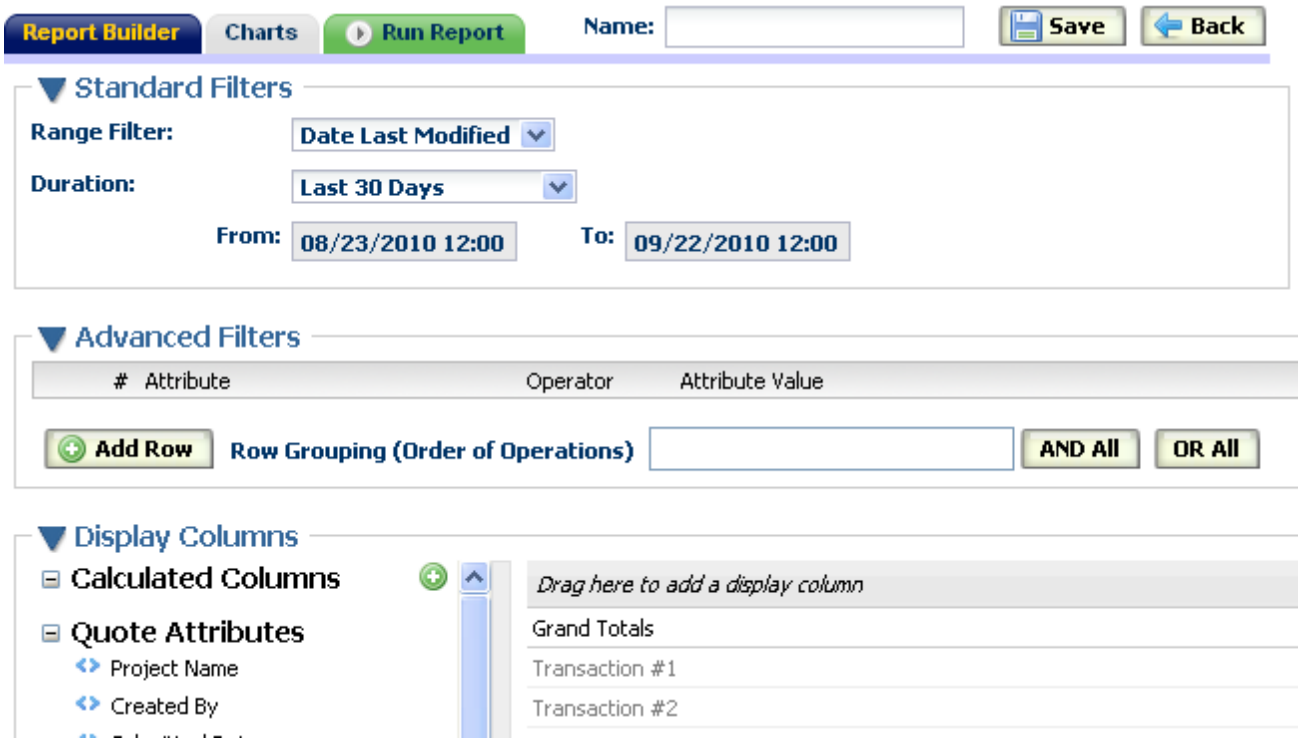


Users can schedule a maximum of 10 reports at a time.

Running a Report

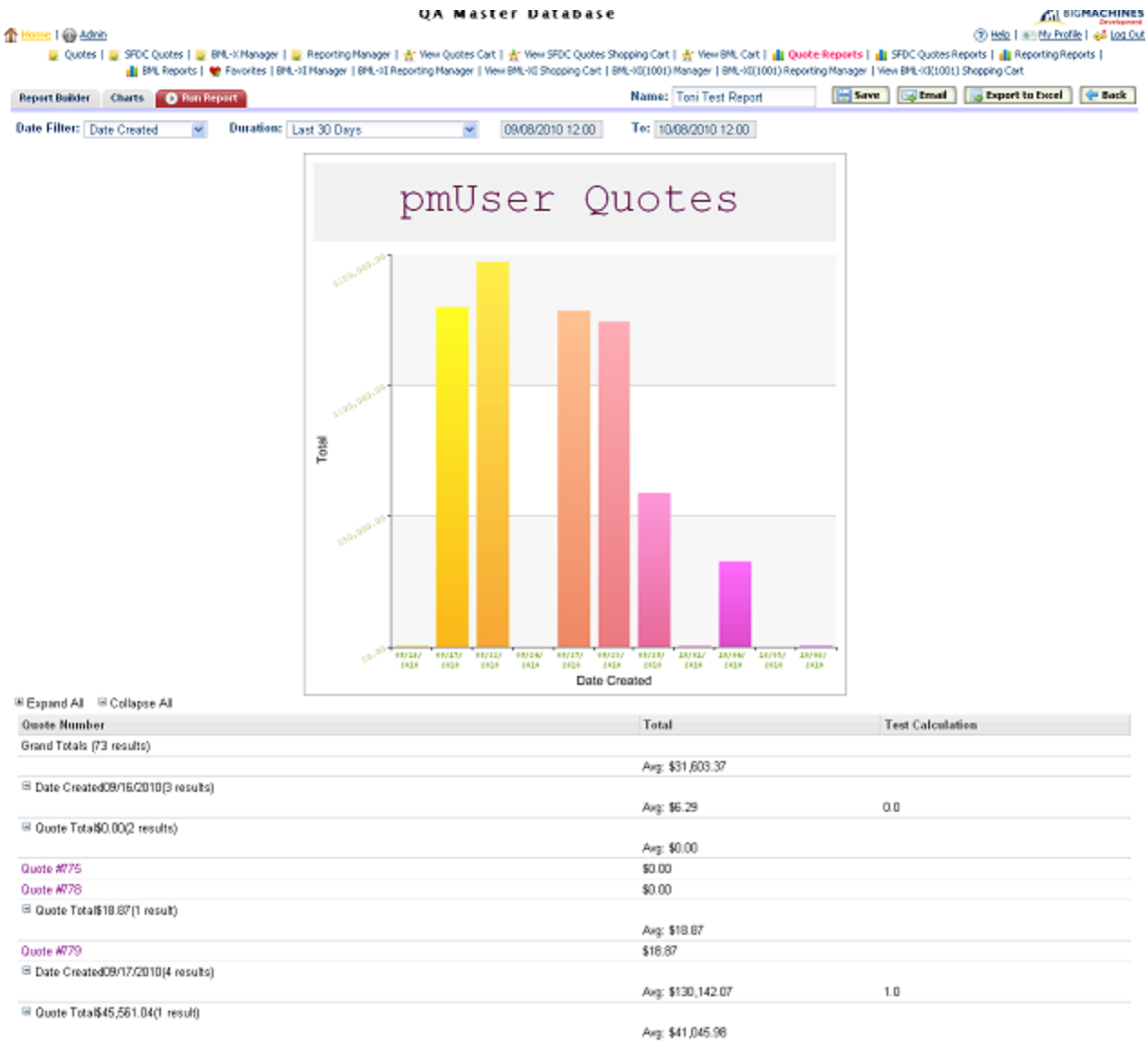
When you have finished creating your reports and charts, you can run the report.

1. Click **Run** from the Report Manager OR if you're in the **Report Builder**, click the Run Report tab in the header bar.



2. Modify the **Standard Filters** by use the drop-down menus for Date Filter (Date Created, Expiration Date, Date Last Modified, etc.) and for Date Duration (Custom, Last 30 Days, Current Quarter, etc.).

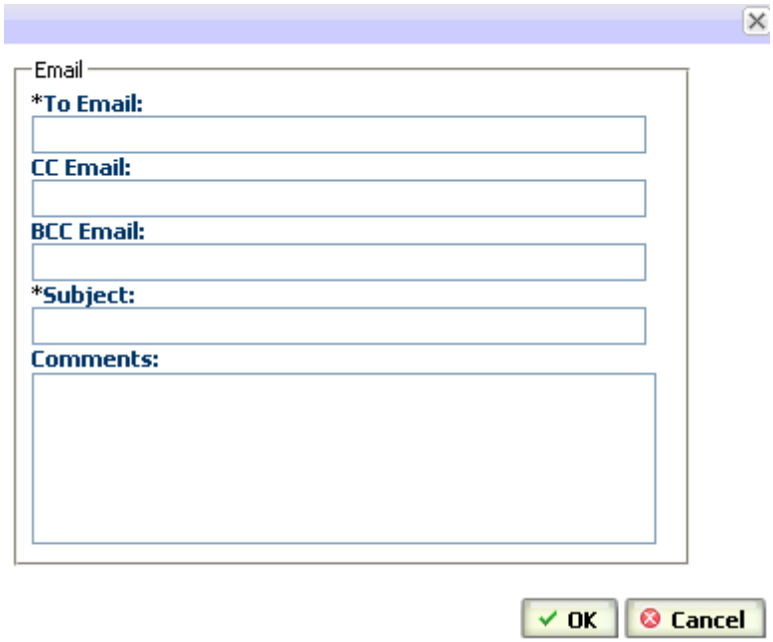
3. Click **Run Report** again to see the report and the chart that you created:



4. If the report that was generated meets your requirements, you can save, email, or export the report to Excel. The report actions are located just next to the report name.



5. Click Email. The following dialog appears:



- Complete the two required fields (marked by an *) and click Send to email your report and chart. The file is sent as an attachment in Excel format.

The user can open transactions when Display Columns are shown as links.

NOTES

Only 2000 records can be shown on the UI. The user must download the results to view more than 2000 results.

Reports can also be run by passing in a blank value as criteria for the filter attributes. In that case, all records are returned.

Reporting is an add-on module and additional charges may apply. Please contact your Project Manager for more details.

The fields that are NOT available for reporting are Composite Summation, Attachment, Rich



Text/description over 255 characters, multi-select menus, Read-only HTML fields and all configurable attributes.

Report results depend on user permissions. If a quote is in a particular step and is hidden from a certain user, that user will not be able to view the transaction in the report results.



If the user does not specify the default criteria, then the report is run on transaction data that has a Date Last Modified within 30 days from the present system date.

Results are organized in groups for display. Pagination occurs at the lowest group level.

RELATED TOPICS

BUILD A REPORT

OVERVIEW

This article assumes you have already created a new report or you are editing a report.

Each Report can have two types of criteria: Standard and Advanced. Results are generated based on transactions that satisfy both of these filter requirements.

The Display Columns section allows the user to build the data that is displayed when the report is run. The user can select attributes to display and specify the way in which the columns must be grouped. The user can choose to display some values that are calculated from numeric/currency attributes that are returned.

ADMINISTRATION

Standard Filters



These filters include the following:

Field	Description
Currency	You can filter by document currency. Reports can return transactions in ALL currencies, as long as the user does not perform any operation, like sum, average, etc. on a Currency type attribute, use that attribute in a calculated column, or use a Currency type attribute in a Group By. In these cases, the user must select a specific currency.
Date Filter	This filter is mandatory. Standard dates that can be used to build reports are Date Created and Date Last Modified. Additionally, users can choose any custom commerce Date attributes that have been mapped to the Data Columns, i.e. Expiration Date.
Duration	This filter is mandatory. You can choose the duration for the date chosen in the Date Filter.

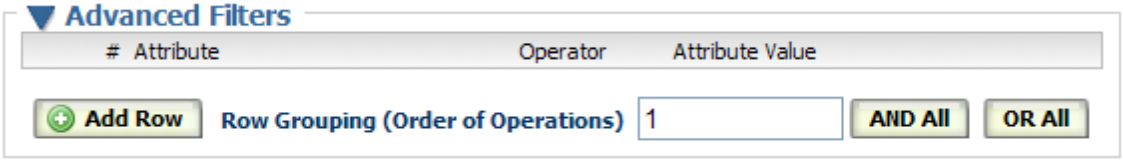


In relation to duration, remember that a week is considered to begin on a Sunday and end on a Saturday. You are allowed to set the beginning of a fiscal years in the Process Administration editor. All Fiscal durations will be based on this date.

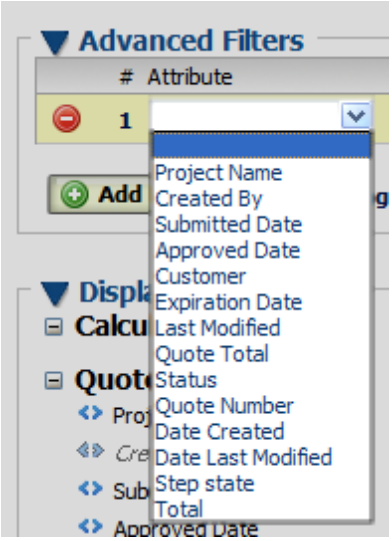
Creating Advanced Filters

The user can define advanced filters on all main-document, sub-document and configurable Text, Numeric, Currency, Date and Menu type attributes that have been mapped to Data Columns.

1. Click **Add Row**.

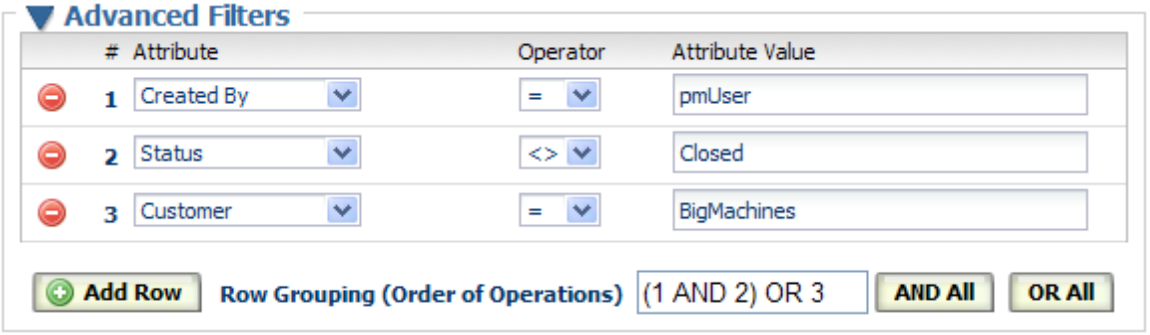


2. Click on the drop-down menu under Attribute and choose the attribute you'd like to filter by.



3. Select the **Operator** and enter an **Attribute Value**.

4. Set the **Row Grouping**, for multiple attributes. If left blank it will default to AND ALL.



Blank values are supported in Filters just as they are supported in saved searches. However, the user cannot create a report without specifying an operator for a filter, unlike in saved searches.

Setting Displayed Columns

Once you have created your standard and advanced filters, you will need to define at least one main-document display column in order to run the report. Failure to do so will result in an error message.

You can select from three options:

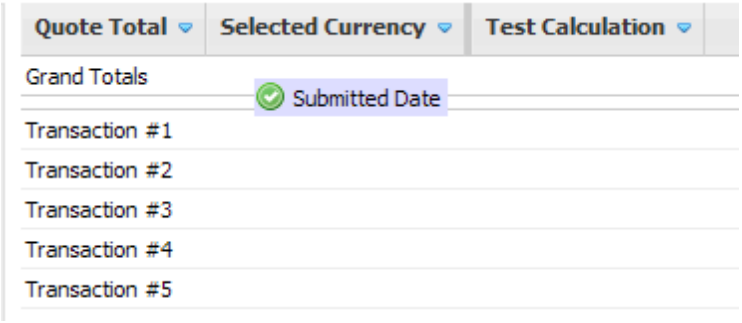
- **Calculated Columns:** This opens a BML-Like editor that allows you to write a script to define the output of the calculated columns.
- **Quote Attributes:** These are the same attributes available in Advanced Filters. All text, numeric, currency and date commerce attributes that are mapped to Data Columns.
- **Line Item Attributes:** These are the line(sub-doc) attributes that are mapped to Data Columns.

---

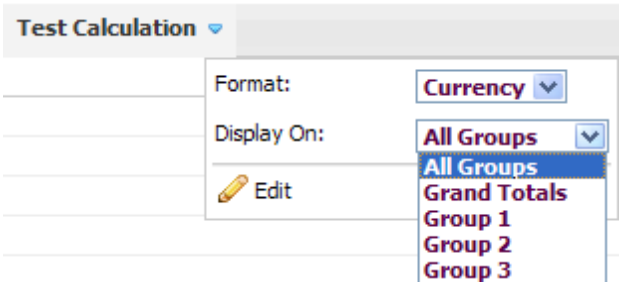
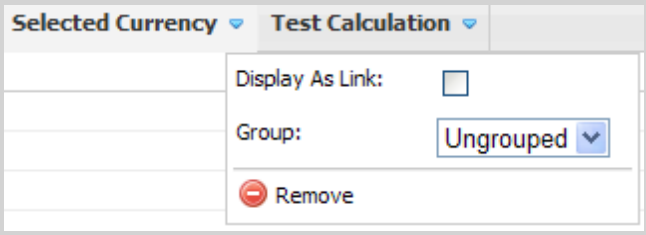
1. Select an attribute to drag-and-drop into the Display Columns panel.
2. A green arrow next to the attribute name means that the attribute can be placed in the position you've selected. A red NO sign will appear next to the attribute if it is not allowed.

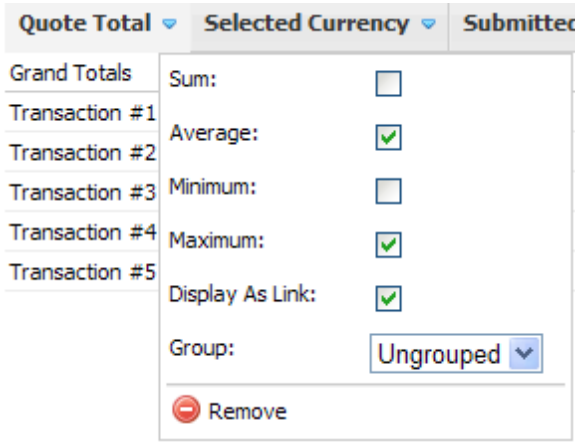


The order you place your columns is how they will appear on the report.

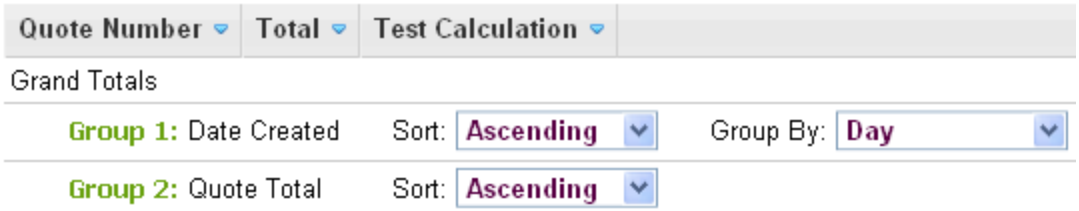


3. Set display column options:

Column Type	What Does It Look Like?	Options	Notes
Calculated Column		• **Format:** Switch between the Number and Currency formats. If the Number format is selected, another option will appear allowing you to set the number of decimal points. • **Display On:** Specify at which group level the column should be displayed. • **Edit:** Make changes to the calculated column.	
Non-Numeric Quote and Line Item Attributes		• **Display as Link**: Displays the attribute as a link that when clicked, takes you to the quote. • **Group:** Determine if this attribute should be used to group the rows. Doing so will remove the attribute as a Display Column. For example, you can group quotes by their Submitted Date or by Selected Currency. • **Remove**: Click to remove the column from the Display Columns panel.	

Column Type	What Does It Look Like?	Options	Notes
Numeric Quote and Line Item Attributes		• **Sum:** Reflect the sum of the aggregate for an attribute. I.e., Quote Totals. • **Average:** Reflect the average for an attribute. • **Minimum:** Reflects the minimum value of all Quote Totals for a group. If there are three quotes ($145,694.23, $34.56, and $35, 434), the minimum will find $34.56. • **Maximum:** Reflects the maximum value of all Quote Totals for a group. If there are three quotes ($145,694.23, $34.56, and $35, 434), the maximum will find $145,694.23.	All numeric quote and line item attributes have the ability to be displayed as aggregates. Aggregates are used in conjunction with Groups to provide data on the x-axis of charts. in this example, you will see the options for the attribute Quote Total.

4. **Group your attributes into rows**. This will remove them from the Display Columns header and place them below.



5. You will then be given the options to sort in Ascending or Descending orders. Depending on the attribute, you may also see options to Group By certain specifics. If you notice the example below, there is a Group By Drop down menu for Group 1 which contains options such as grouping by day, week, month, quarter, year and more.

NOTES



Grouping becomes very important if you plan on adding charts to your report.



Pagination of reports occurs at the group level.

Results within a group will be sorted in ascending or descending order for text, menu, currency and numerical attributes. Additionally, by Duration for date attributes.



All commerce attributes used in reporting must be mapped to Data Columns. HTML, RTE, File Attachment, composite attributes, history fields and approval-related attributes are not available for reporting.

RELATED TOPICS

REPORTING LIMITS

There are few limits to be aware of when using Native Reporting.

1. There's a setting, General Site Options à Row Counter Per Page, that allows the fullaccessuser to set the number of results per page. The maximum is 501. If the number of rows returned is greater than the value set here, pagination occurs. If the report contains a bunch of grouping, this pagination will be applied to each group that returns rows in excess of 501—or whatever limit the admin set.

2. The limit here is a 'feature' of Excel. Older versions of Excel had a limit of ~65K per worksheet. Excel 2007 has a limit of ~1M (1,048,576 actually).

3. There are no limits on Grouping.

RELATED TOPICS

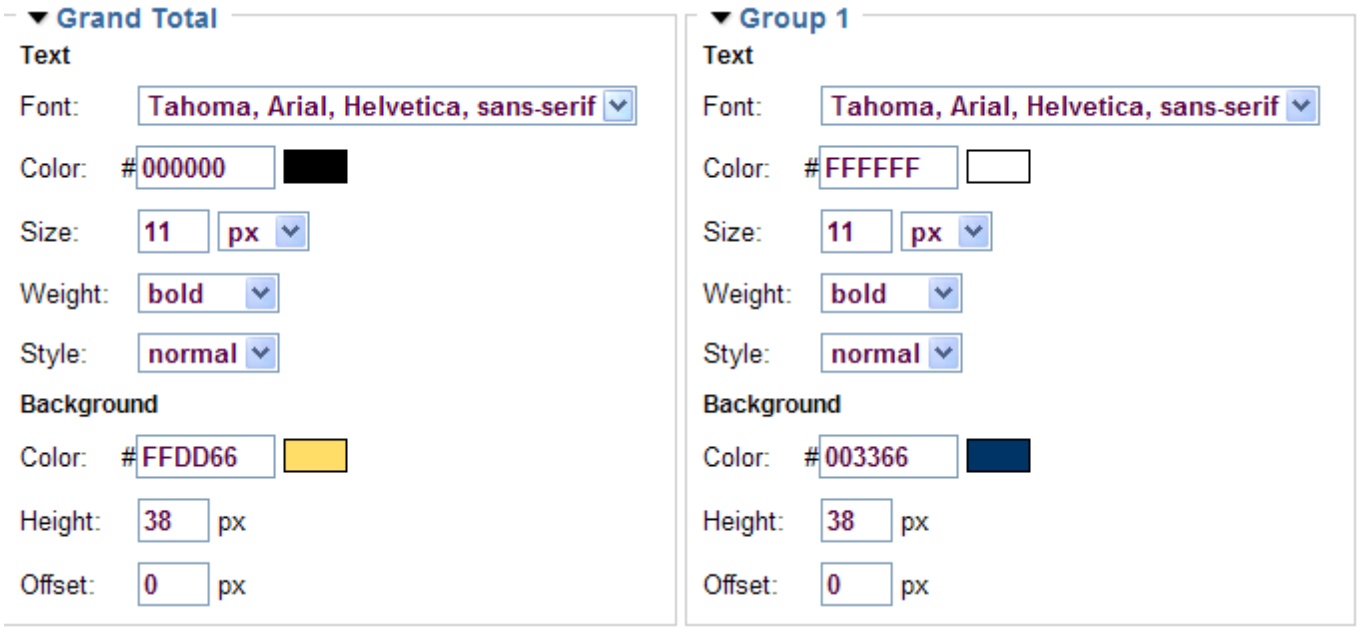
STYLESHEET FOR REPORTS

OVERVIEW

As seen on the page, you can make settings for displaying options for multiple groups. Standard Navigation Path:Home Page > Report Manager in the navigation pane > Report Stylesheet.

As seen on the page, you can make settings for displaying options for multiple groups.

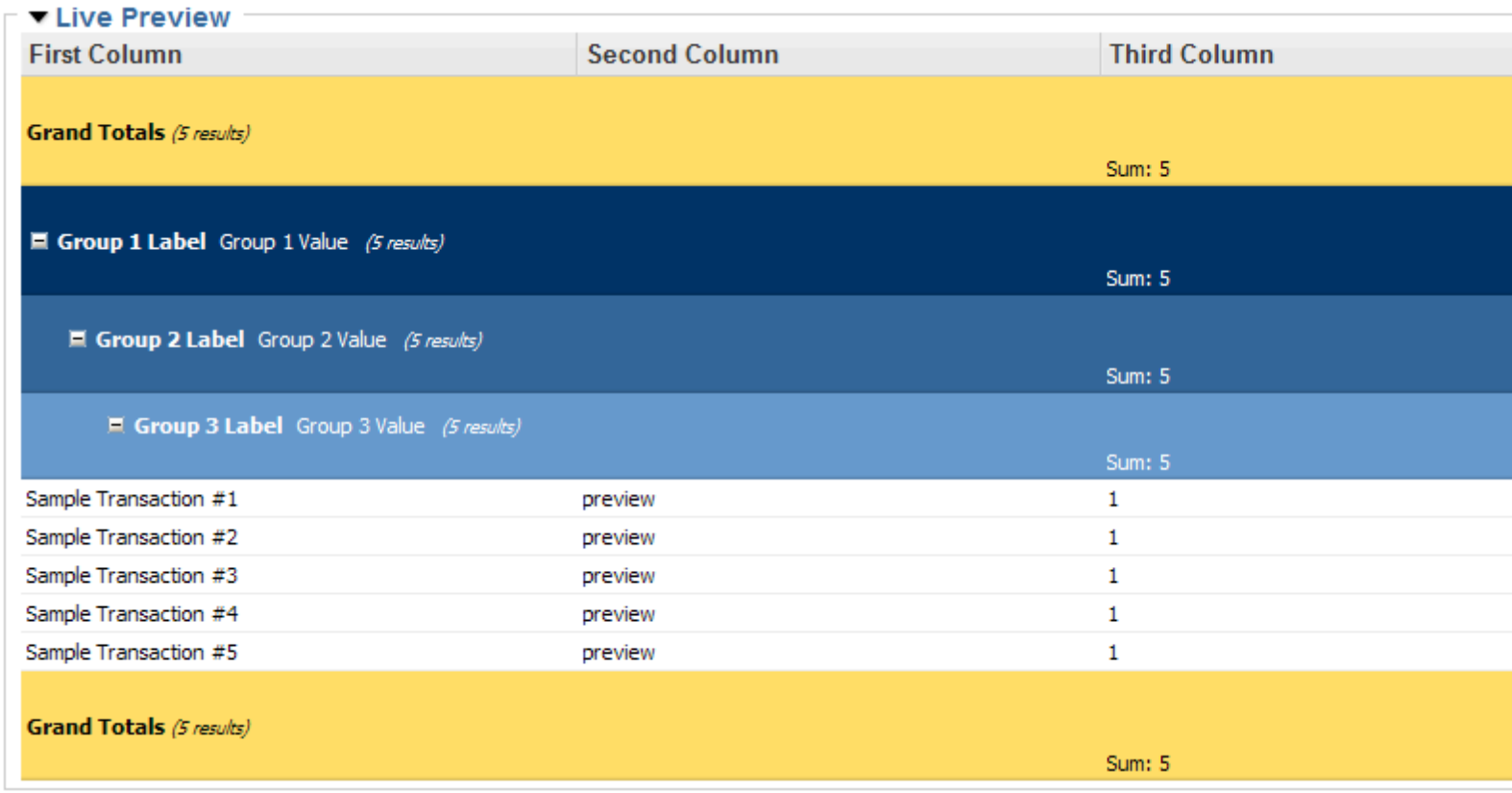
StyleSheet Editor



Property	Description
Font Family	Choose the font family of the text from the drop-down menu. The default is Tahoma, Arial, Helvetica, sans-serif.
Font Color	Specify the color of the font by entering a HEX value or by clicking on the color indicator to open a color selector.
Font Size	Choose the font size of the text from the drop-down menu.
Style	Choose normal or italic from the drop-down menu.
Background Color	Specify the background color of the label by entering a HEX value or by clicking on the color indicator to open a color selector.
Background Height	Specify the height of the label, in pixels.
Background Offset	Specify how far to the right that the label should be indented, or offset, in pixels.

Live Preview

You can see a live preview of the changes you have made in the section below the StyleSheet Editor.

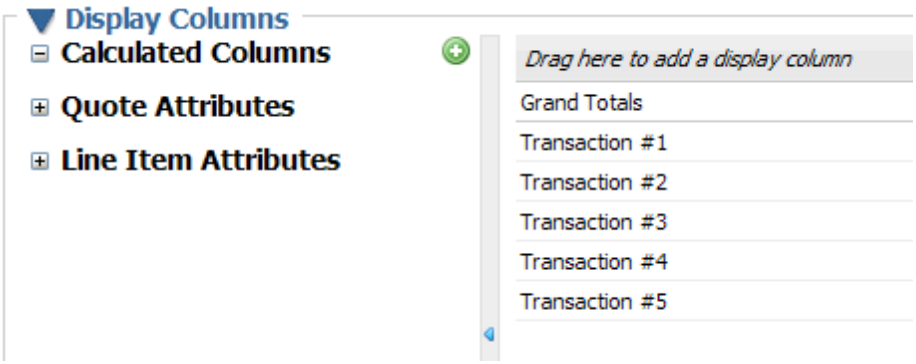


RELATED TOPICS

CALCULATED COLUMNS

OVERVIEW

You can create your own calculated columns within the Report Builder. To create a calculated column, click the Add icon next to the Calculated Columns label found under Display Columns.



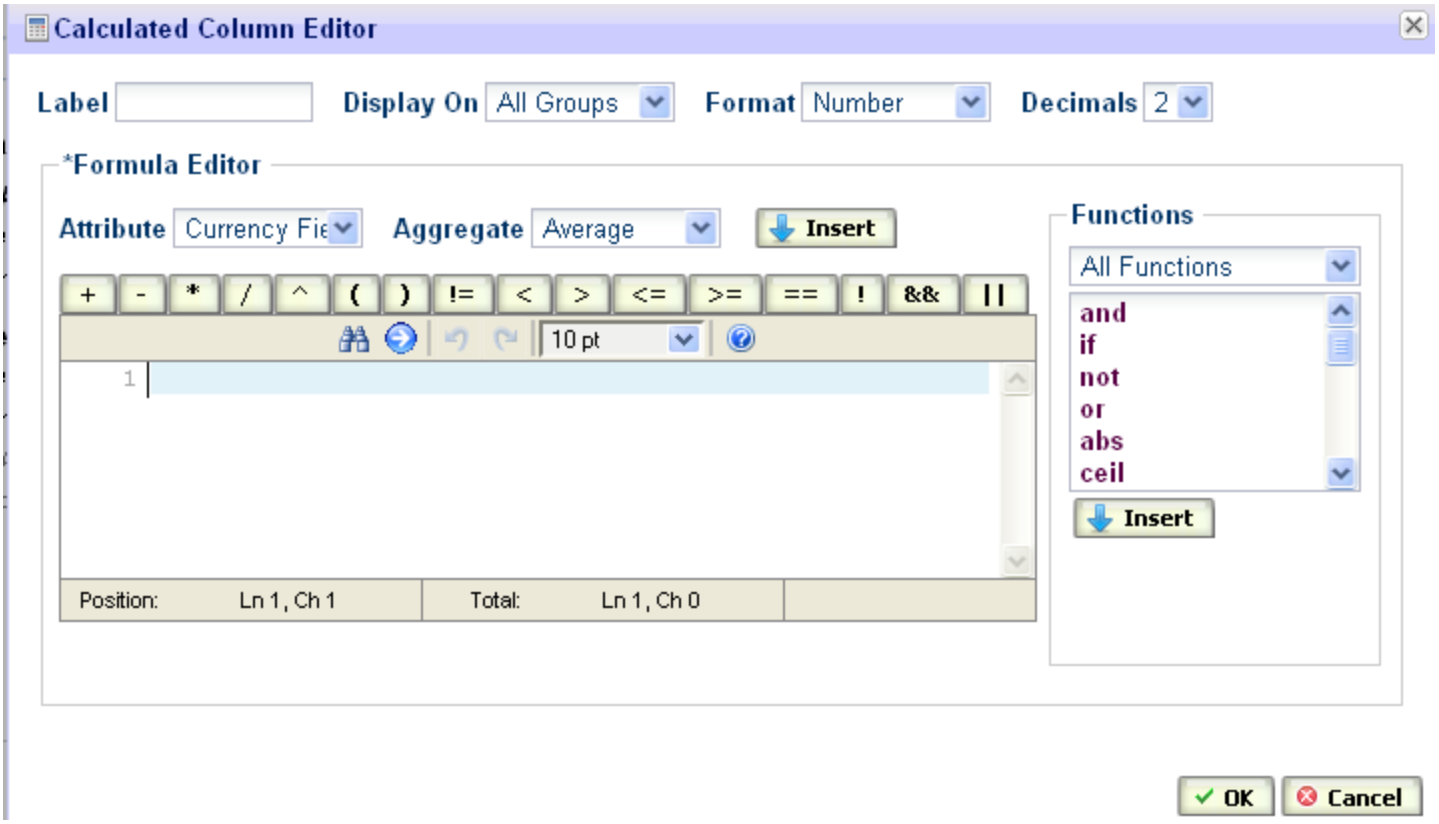
ADMINISTRATION

Creating a Calculated Column

1. Enter a **Label name** on the Calculated Column Editor.
2. Determine if the calculation should be displayed next to a particular group, All Groups or Group Totals.
3. Determine the **Format of the column:** Number or Currency.



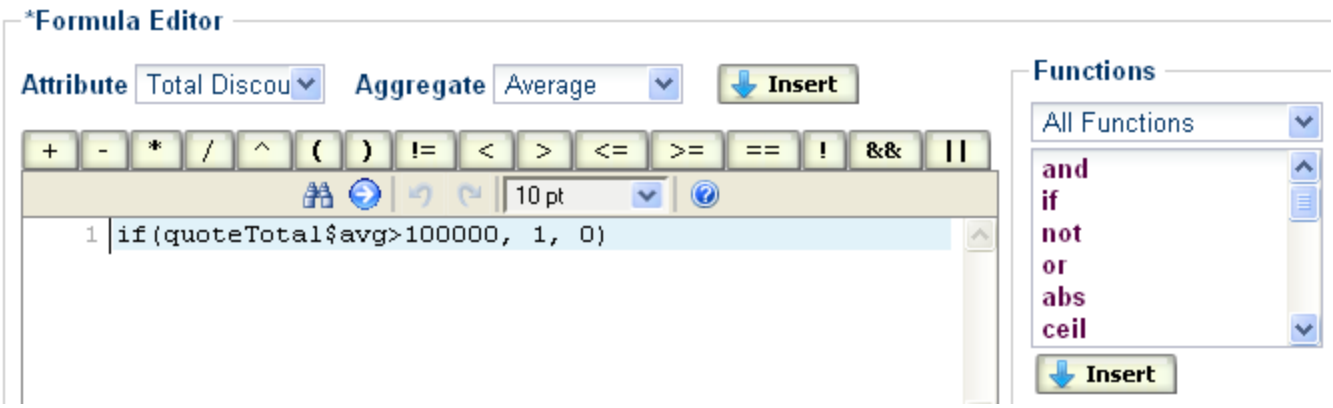
If you choose number, make sure to select between 0-4 allowed decimal places.



4. Choose the **attribute(s)** that you would like to include in your calculation. *This is found in the Formula Editor section.*

> The attributes that are in this list are the numeric or currency attributes that you have mapped to Data Columns.

5. Choose the **aggregate** that you will use to calculate the column (i.e.: Average, Count, Sum, Minimum, Maximum).
6. Click the **Insert button** to add the attribute and aggregate to the script editor.
7. Use the **operator buttons** above the script editor to help create your calculation.
8. Choose a Logical, Mathematical or Summary function to add to the calculation from the drop-down menu and **click Insert**.

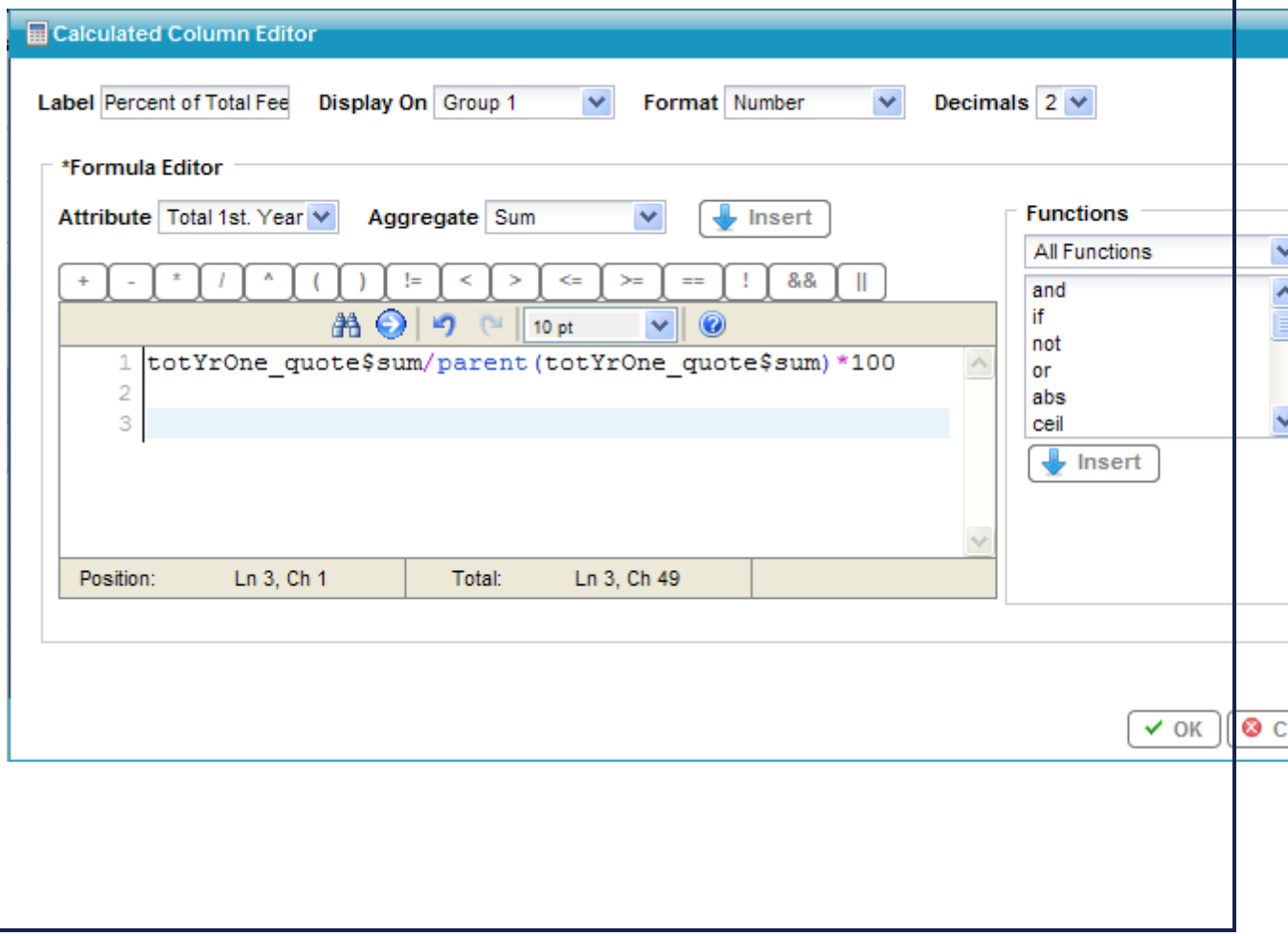


9. Click **OK** when you have finished creating the calculation to save it.

USE CASES

Use Case: Percentage of Total Fees

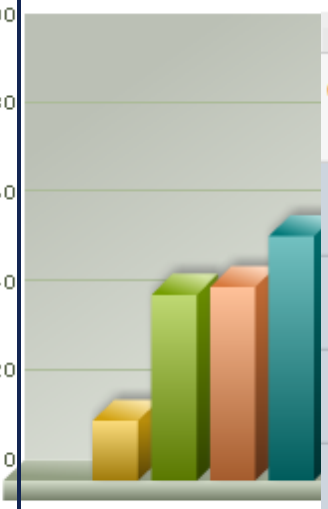
Quote Number	Expiration Date	Total 1st. Year Fees	Percent of Total Fees
Grand Totals *(80 results)*		Sum: $964,712.07	
Status Finalized *(16 results)*		Sum: $212,751.29	22.05
Rep Name Jim Halpert *(15 results)*		Sum: $183,258.13	
Rep Name Product Management *(1 result)*		Sum: $29,493.16	
Status Order Submitted *(7 results)*		Sum: $64,234.69	6.66
Rep Name Jim Halpert *(6 results)*		Sum: $56,694.95	
Rep Name Product Management *(1 result)*		Sum: $7,539.74	
Status Pending *(52 results)*		Sum: $629,695.34	65.27
Rep Name Jim Halpert *(31 results)*		Sum: $391,053.74	
Rep Name Product Management *(21 results)*		Sum: $238,641.60	
Status Under Review *(5 results)*		Sum: $58,030.75	6.02
Rep Name Hugh Thompson *(1 result)*		Sum: $6,493.49	
Rep Name Jim Halpert *(4 results)*		Sum: $51,537.26	

What Am I Doin-g?	How Do I Do It?	What Does it Look Like?
Fees based on Quote Status (Group 1)	In this example, we'd like to display the per-centage of the attrib-ute "Total First Year Fee's" for each quote status. Since the quote status is in Group 1, this is where we'd like this cal-culation to be dis-played. Notice that we are dividing the Total 1st Year Fees by the Parent level of Total first Year Fees and then multiplying by 100 to get a per-centage.	

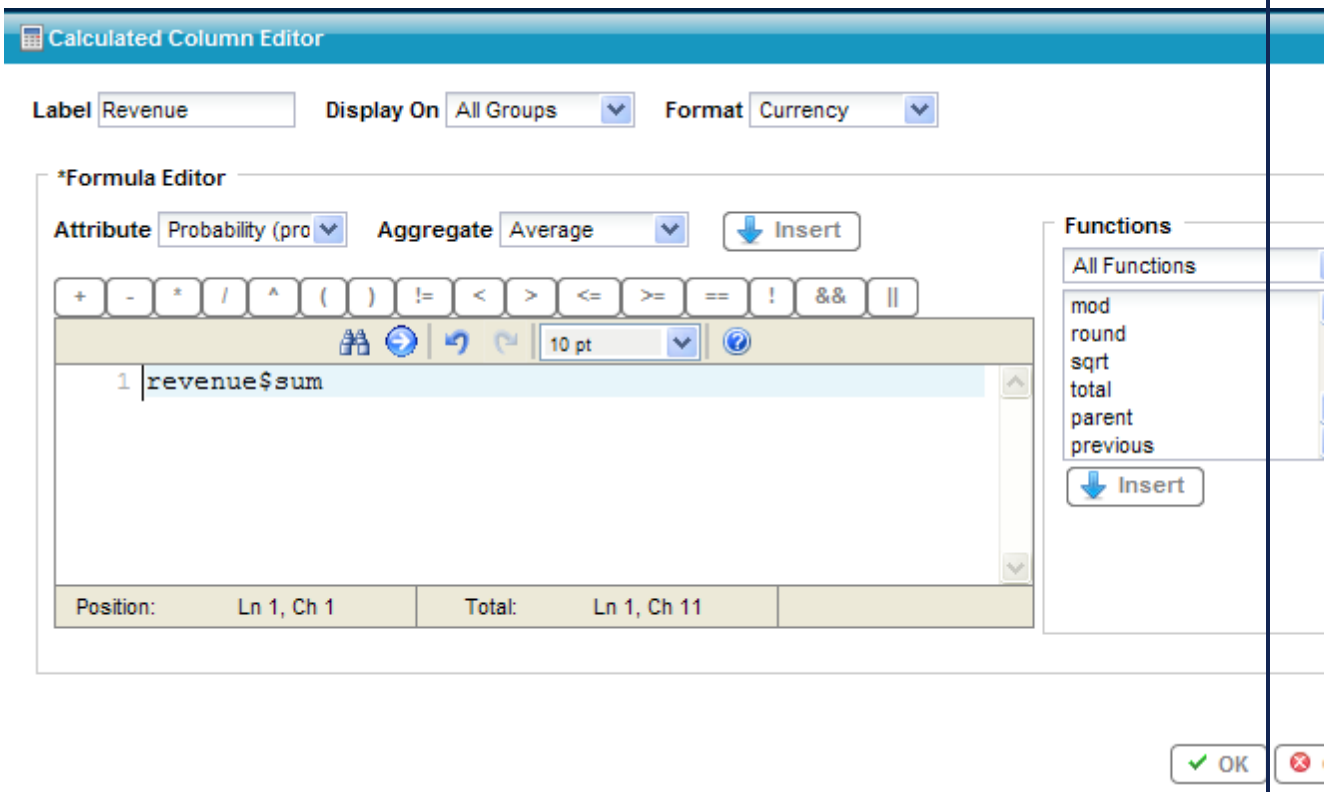
What Am I Doin-g?	How Do I Do It?	What Does it Look Like?
Based on Rep Name (Group 2)	Let's say you wanted to see the per-centage of total first year fee's based on Rep Name. Note that the rep name is in Group 2, a sub-level of Group 1 (Status). Since Status is the "par-ent" of Rep Name, you'll notice that the per-centages shown by the rep name add up to 100% of each Status.	**Status** Finalized *(16 results)* Sum: $212,751.29 **Rep Name** Jim Halpert *(15 results)* Sum: $183,258.13 86.14 HiTech-43316 02/13/2011 $27,821.74 HiTech-36186 11/24/2010 $15,965.96 HiTech-40164 01/01/2011 $8,882.43 HiTech-37864 12/09/2010 $8,080.30 HiTech-37877 12/09/2010 $7,993.11 HiTech-39443 12/24/2010 $11,415.49 HiTech-37854 12/09/2010 $8,311.05 HiTech-37858 12/09/2010 $11,272.41 HiTech-37871 12/09/2010 $11,619.35 HiTech-37883 12/09/2010 $14,474.05 HiTech-36156 11/24/2010 $9,979.23 HiTech-41590 01/15/2011 $5,561.61 HiTech-37889 12/09/2010 $7,539.74 HiTech-37862 12/09/2010 $9,236.55 HiTech-40630 01/06/2011 $25,105.11 **Rep Name** Product Management *(1 result)* Sum: $29,493.16 13.86

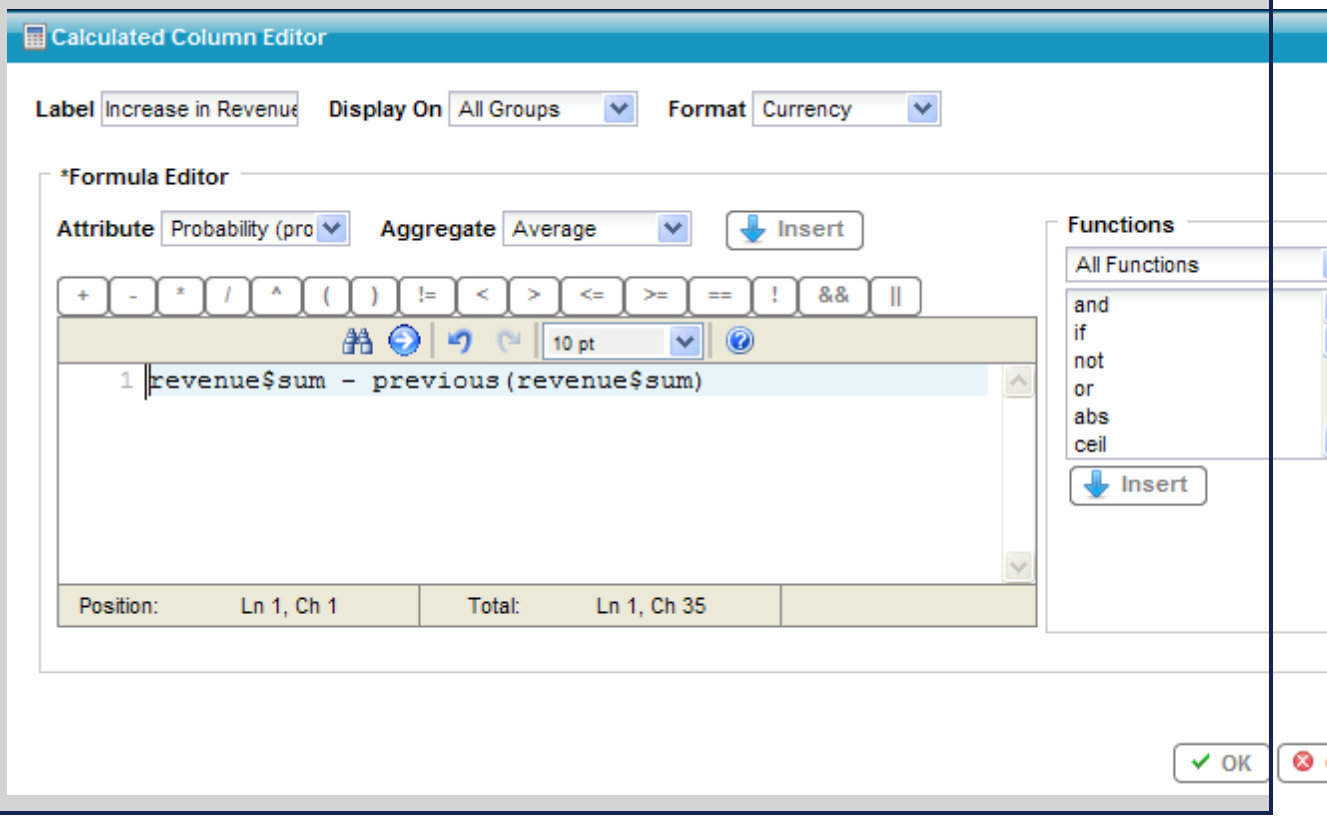
Use Case: Changes in Revenue

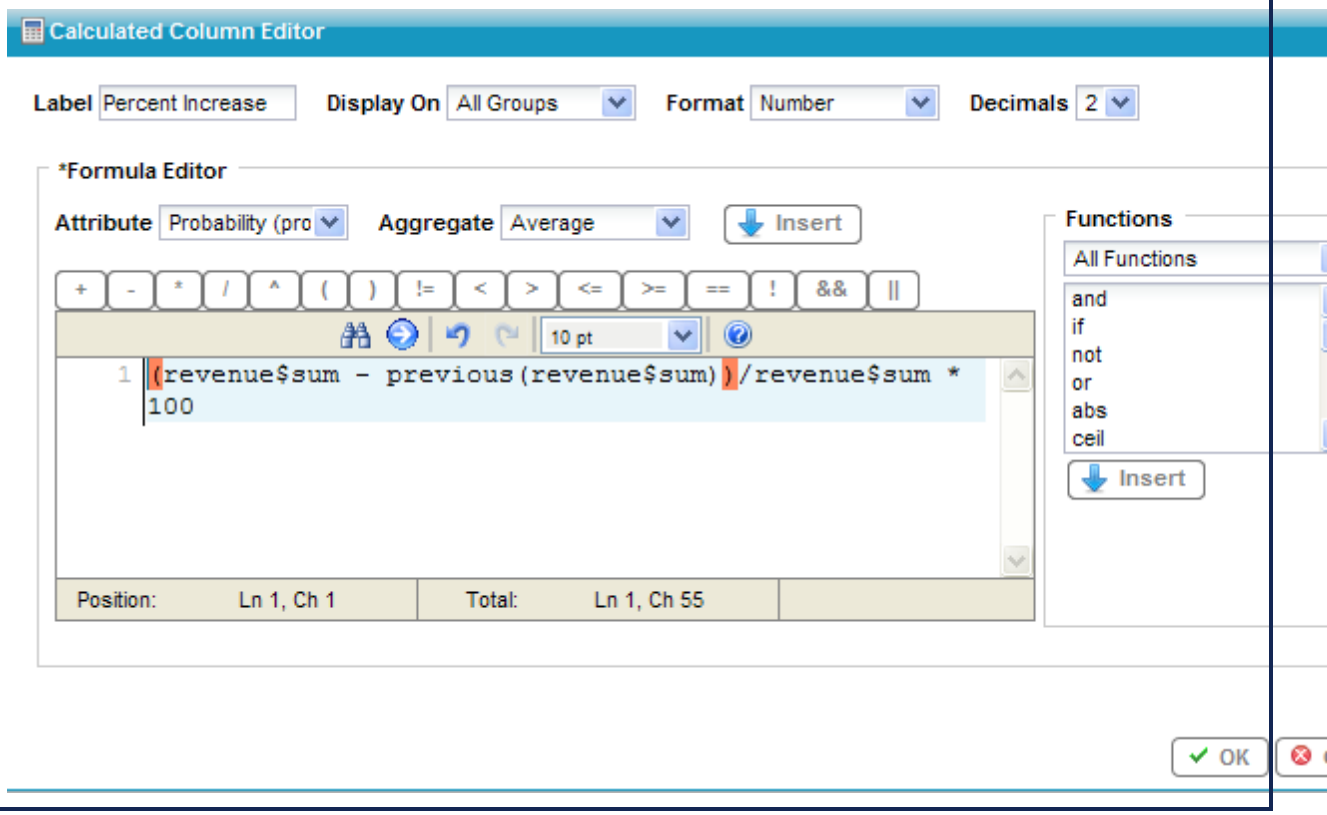
In this example, we'd like our report to display the total annual revenue, increase in revenue and percentage of that change.



Quote Number	Company	Revenue	Increase in Revenue	Perc
Grand Totals *(1835 results)*		$101,107,762,642.42		
⊞ **Win Date** 01/01/2000 *(184 results)*		$27,307.44		
⊞ **Win Date** 01/01/2001 *(174 results)*		$31,540.98	$4,233.54	13.42
⊞ **Win Date** 01/01/2002 *(181 results)*		$54,095.47	$22,554.49	41.69
⊞ **Win Date** 01/01/2003 *(194 results)*		$95,593.50	$41,498.03	43.41
⊟ **Win Date** 01/01/2004 *(193 results)*		$211,649.59	$116,056.09	54.83

What Am I Doing?	How Do I Do It?	What Does it Look Like?
Calculating Revenue	The first column that is displayed is a calculated column named Revenue. This should be displayed on All Groups level and we want it to display as a Currency. We will select the attribute "Revenue" and the aggregate "Sum" and insert it into the script editor. This will give us the sum of all revenue for all groups.	

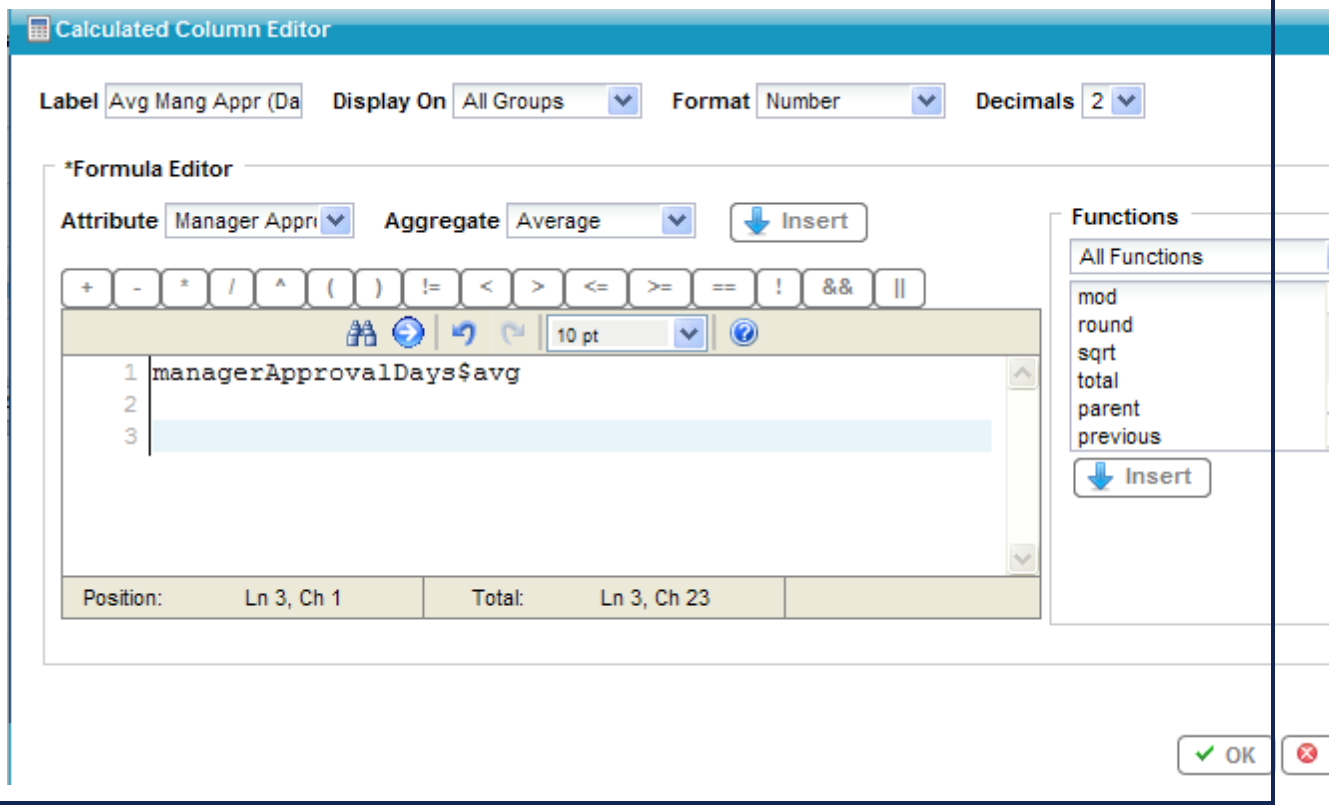
What Am I Doing?	How Do I Do It?	What Does it Look Like?
Calculating Increase in Revenue ($)	To calculate the increase in revenue, we'll take the calculation from above and subtract the previous year's revenue total by using the 'previous' function. Make sure the number Format is 'Currency'.	

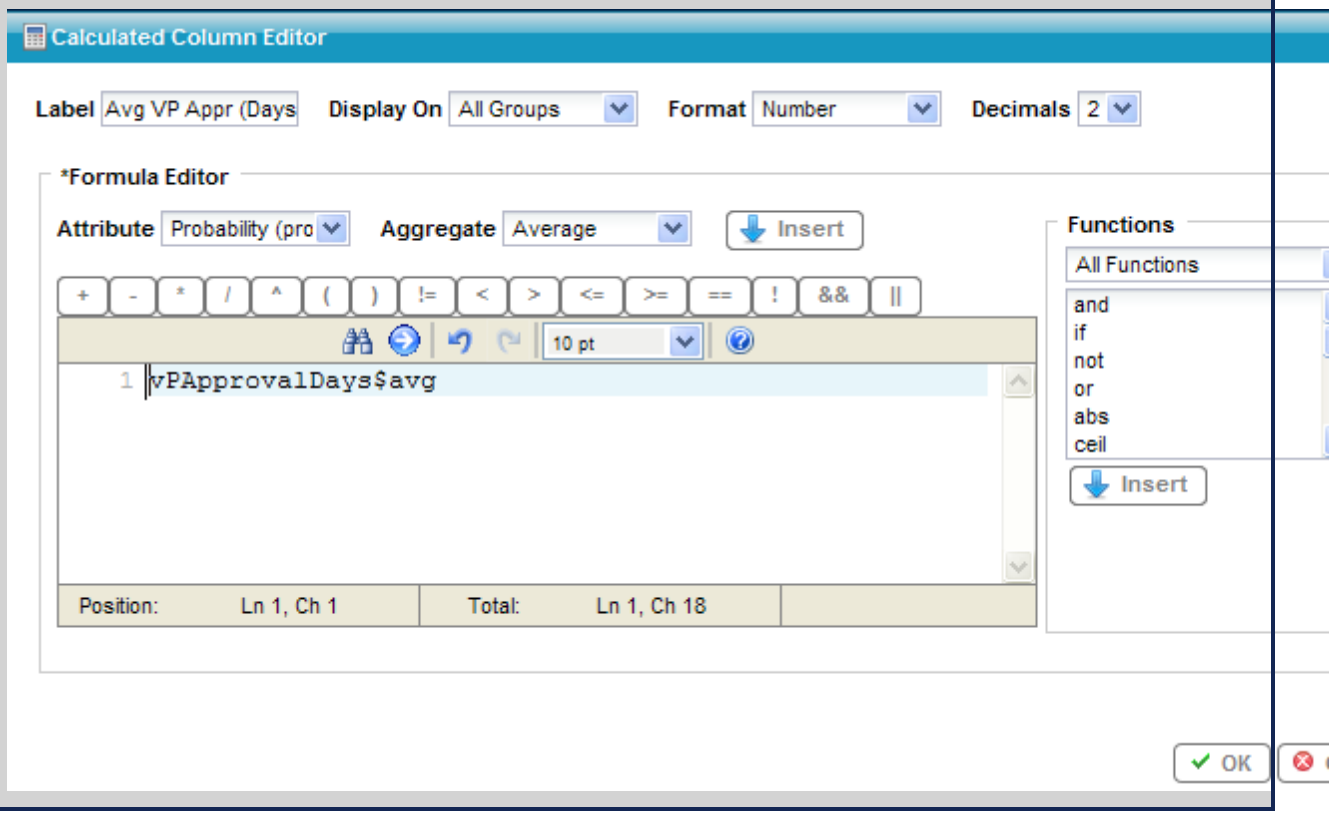
What Am I Doing?	How Do I Do It?	What Does it Look Like?
Calculating Increase in Revenue (%)	To complete the report, we'll take the calculation from above and divide it by sum of all revenue and multiply it be 100 to get a percentage. Note: You should make sure that the Number format is used here.	

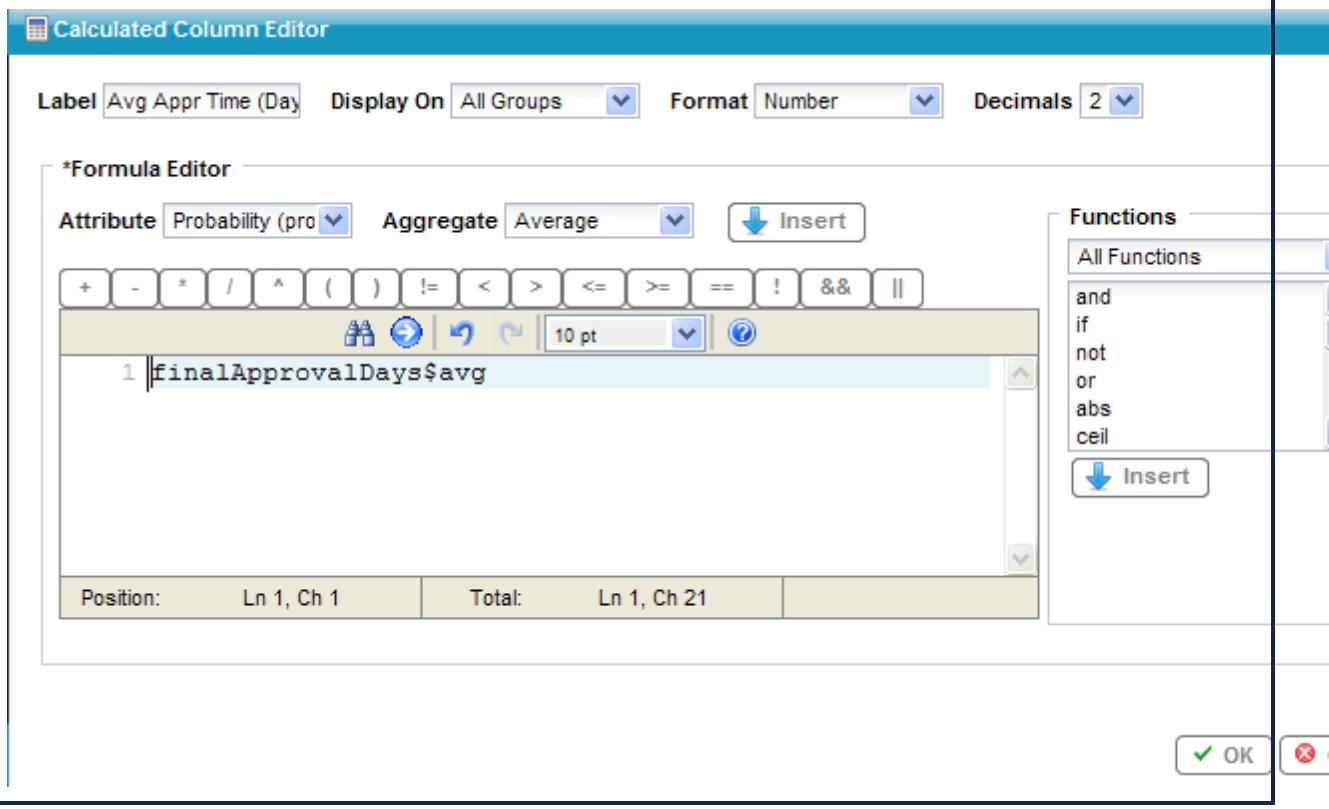
Use Case: Average Time Between Approvals

You can use calculated columns to create report that shows the average time it takes for a qutoe to be approved. In this case, the chart is displaying the results of the average days it takes a manager to approve a quote based on who the sales rep is.

Quote Number	Manager Approval Date	VP Approval Date	Total 1st. Year Fees	Avg Mang Appr (Days)	Avg
Grand Totals *(44 results)*			Avg: $188,633.52	61.0	244
Rep Name Bobby Monahon *(1 result)*			Avg: $95,315.00	45.0	180.
HiTech-16421	07/17/2010	01/13/2011	$95,315.00		
Rep Name Dennis Adams *(2 results)*			Avg: $424,309.40	51.5	206.
HiTech-38753	07/07/2010	01/23/2011	$744,100.50		
HiTech-42732	07/01/2010	01/29/2011	$104,518.30		
Rep Name Hugh Thompson (JP) *(1 result)*			Avg: $93,523.00	58.0	232.
HiTech-28414	06/21/2010	02/08/2011	$93,523.00		
Rep Name Jason Danielson *(1 result)*			Avg: $2,573,932.50	46.0	186.
HiTech-24885	07/14/2010	01/16/2011	$2,573,932.50		
Rep Name Rob Mullany *(1 result)*			Avg: $333,826.00	44.0	178.
HiTech-4042	07/18/2010	01/12/2011	$333,826.00		
Rep Name Ryan Gauthier *(1 result)*			Avg: $89,221.50	58.0	234.
HiTech-5926	06/20/2010	02/09/2011	$89,221.50		
Rep Name SF User *(12 results)*			Avg: $80,562.73	67.67	271.
SFDC-12661	07/01/2010	01/29/2011	$75,587.80		
SFDC-20208	05/28/2010	03/04/2011	$81,538.00		
SFDC-20323	05/16/2010	03/16/2011	$81,538.00		
SFDC-20338	05/10/2010	03/22/2011	$81,538.00		
SFDC-20347	05/17/2010	03/15/2011	$81,538.00		

Approva-l	How Do I D-o It?	What Does it Look Like?
Average Manager Approval	Select the attrib-ute "Man-ager Approval Days" from the Attribute drop down menu and select the Aggreg-ate "Aver-age".	

Approva-l	How Do I D-o It?	What Does it Look Like?
Average VP Appro-val	Select VP Approval Days from the Attribute drop-down menu and select the Aggreg-ate "Aver-age".	

Approva-l	How Do I D-o It?	What Does it Look Like?
Average Final Approval	Select "Final Approval Days" from the Attribute drop-down menu and select the Aggreg-ate "Aver-age".	

NOTES

Remember that attributes must be mapped to data columns in order to be used as attributes in Cal-culated Columns.

Calculated columns will be displayed on the right-edge of the displayed columns in the order in which they are created.

RELATED TOPICS

CHARTS

OVERVIEW

Once you have created your report using the Report Builder, you can either run the report as is or you can add graphic representation in the form of charts. This can only be done if the user has specified aggregates for numeric and/or currency values. An Aggregate and Group By are required (one for the x-axis and one for the y-axis).

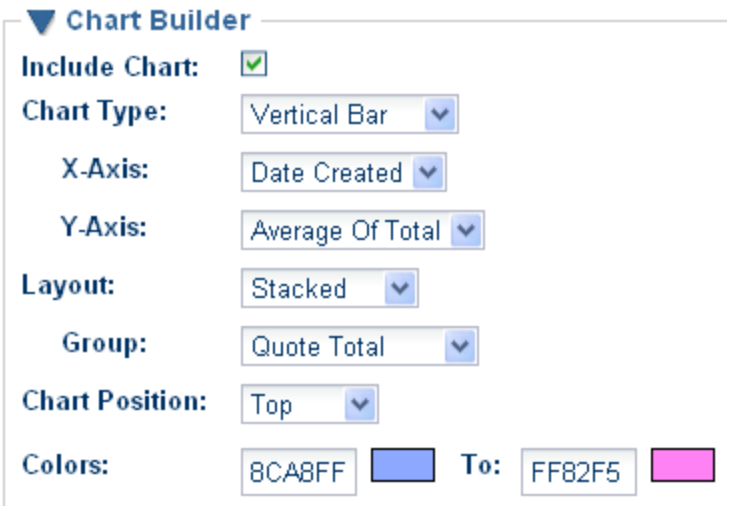
There are multiple chart forms available: Vertical Bar, Horizontal Bar, Line, Pie and Gauge. You can use a Chart Builder, Additional Settings and Preview section to assist you in the creation of visually appealing graphics.

Vertical Bar, Horizontal Bar and Line Charts

The vertical bar, horizontal bar and line charts are used as graphical representations of data using bars, columns or lines to compare data. They all share the same Chart Builder and Additional Settings options.

Chart Builder

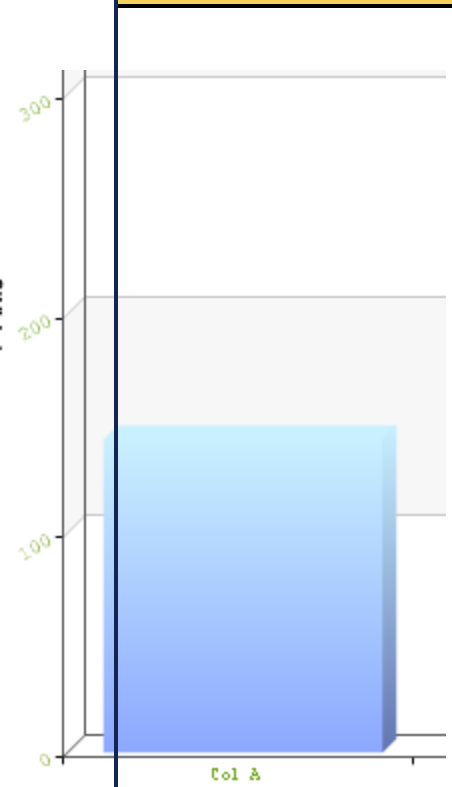
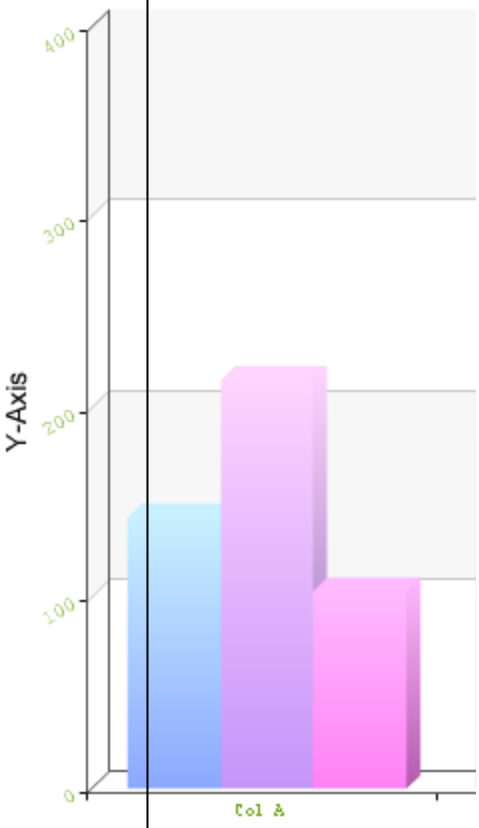
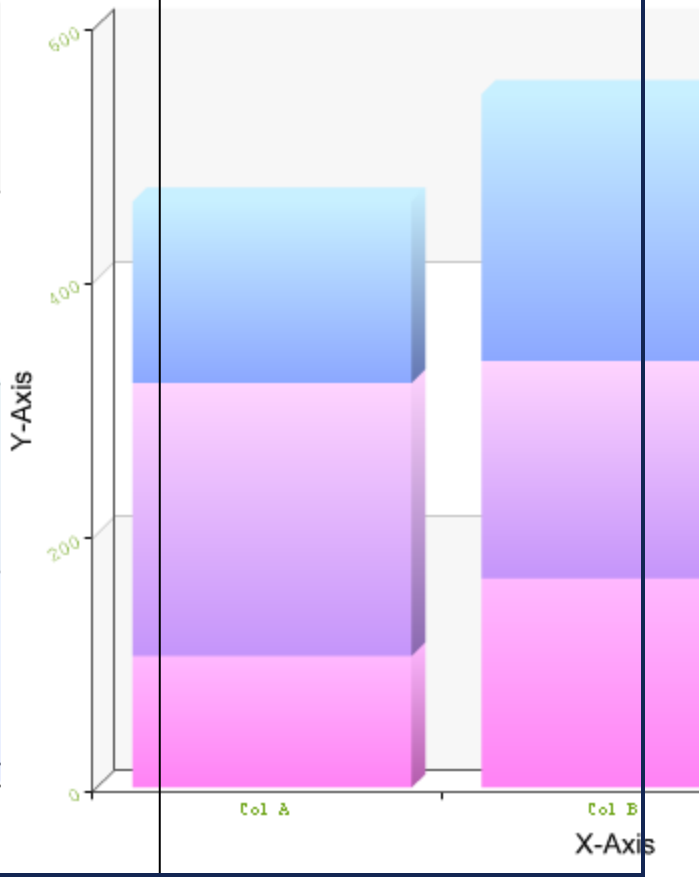
Chart Builder is a section where you select if you'd like to include a chart, which kind of chart you'd like to show and actually build the chart based on the attributes you selected in the Report Builder.



Option	Description
Include Chart	Click the checkbox to include a chart with your report.
Chart Type	Use the drop-down menu to select the pie chart.
X-Axis and Y-Axis	Use the drop-down menu to select which attribute will be represented on the x-axis and y-axis. This selection will also be the axis name on the chart. If you have grouped your attributes in rows, these will be available for use.
Layout	Select from Ungrouped, Grouped or Stacked.
Group	Select how the chart will be grouped.
Chart Position	Determine if you'd like the chart to be displayed on the top or the bottom of the report.

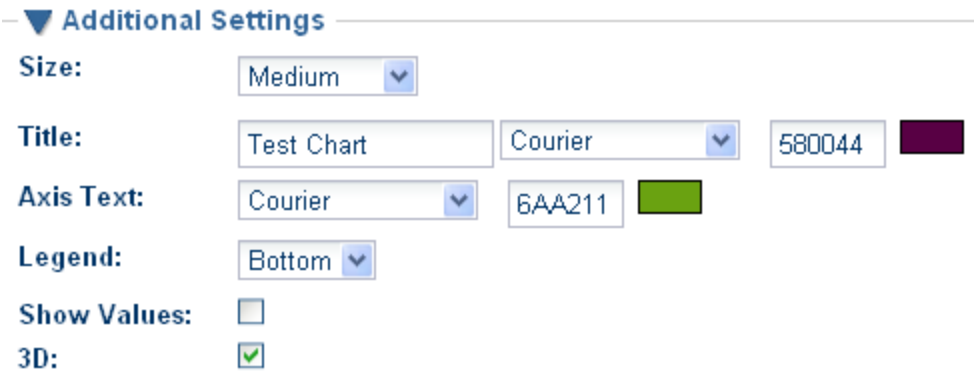
Option	Description
Colors	Select colors to represent the spectrum of data on the x-axis and y-axis. The first value you select will represent the first piece of data and the second color will represent th e final piece of data on the report. The system automatically selects comparable colors for the data in between the two.

Layout Options

Ungrouped	Grouped	Stacked
		

Additional Settings

Additional Settings is the section where you can select chart options, such as the size, the title, the axis text and if you'd like to include a legend.



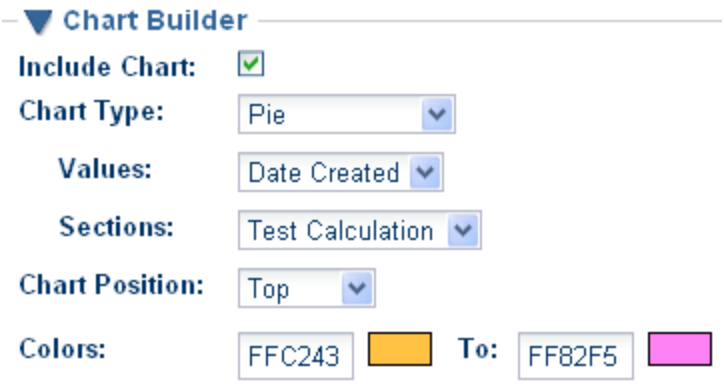
Option	Description
Size	Select the display size of the chart from the drop-down menu: Small, Medium, Large, or Extra-Large.
Title	Enter the chart title in the Title text field.
Title Font	Select the font you'd like the title to be displayed in from the drop-down menu: Courier, Helvetica, Times New Roman, or Arial Unicode MS.
Title Color	Enter a HEX (hexi-decimal) color code into the text box, or click on the color to open a color selector. Use this to select the appropriate color and gradient for the title text. 
Axis Text	Select the font you'd like the axes to be displayed in from the drop-down menu: Courier, Helvetica, Times New Roman or Arial Unicode MS.
Axis Color	Enter a HEX (hexi-decimal) color code into the text box, or click on the color to open a color selector. Use this to select the appropriate color and gradient for the axis text.
Legend	Determine if you'd like to display a legend. If so, use the menu to decide if it should be below the chart or to the side of it.
Show Values	Select the checkbox if you'd like to display the data labels directly on the chart.
3D	Select the checkbox if you'd like to display the chart in 3D. This option is not available for the line graph.

Pie Charts

Pie charts work well when you are trying to graphically represent the proportion of values as they add up to the total. For example, if you'd like to see quote totals from the last 30 days, you can see the entire total represented and which quotes make up the largest and smallest 'pieces of the pie'.

Chart Builder

There are a few differences between the Chart Builder shown for Vertical Bar graphs and Pie Charts.



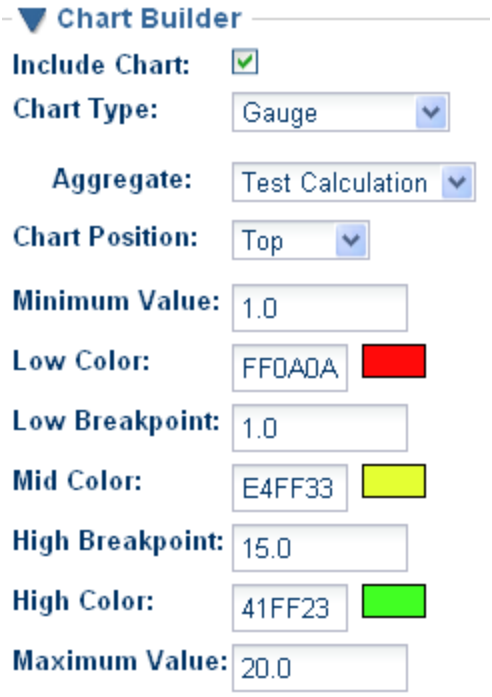
Option	Description
Include Chart	Click the checkbox to include a chart with your report.
Chart Type	Use the drop-down menu to select the pie chart.
Selections	Use the drop-down menu to select which attribute will be represented on the y-axis. This selection will also be the axis name on the chart. Aggregates and calculated columns will be available for use.
Values	Use the drop-down menu to select which attribute will be represented on the x-axis. This selection will also be the axis name on the chart. If you have grouped your attributes in rows, these will be available for use.
Chart Position	Determine if you'd like the chart to be displayed on the top or the bottom of the report.
Colors	Select colors to represent the spectrum of data on the x-axis and y-axis. The first value you select will represent the first piece of data and the second color will represent th e final piece of data on the report. The system automatically selects comparable colors for the data in between the two.

Gauge Charts

Gauges are used to track key performace indicators for a business to measure success. They are often used at the executive level and are frequently used in 'dashboards'. Gauges are often set with a target or a range in mind. The black needle or the dial indicates the current value of the attribute (or key performance indicators). For example, you can set a sales goal for the year, then you can also set mid-point ranges to represent sales targets by quarter.

Chart Builder

There are many new options available to you when creating a gauge.



Option	Description
Include Chart	Click the checkbox to include a chart with your report.
Chart Type	Use the drop-down menu to select the gauge chart.
Aggregate	Use the drop-down menu to select which attribute will be represented on the x-axis. This selection will also be the axis name on the chart. If you have grouped your attributes in rows, these will be available for use.
Chart Position	Determine if you'd like the chart to be displayed on the top or the bottom of the report.
Minimum Value	Set the minimum value for the attribute.
Low Color	Enter a HEX color code into the text box, or click on the color to open a color selector. Use thsi to select the appropriate color and gradient for the low range.
Low Breakpoint	Set the range breakpoint.
Mid Color	Enter a HEX color code into the text box, or click on the color to open a color selector. Use thsi to select the appropriate color and gradient for the middle range.
Hi Breakpoint	Set the second range breakpoint.
High Color	Enter a HEX color code into the text box, or click on the color to open a color selector. Use thsi to select the appropriate color and gradient for the high range.
Maximum Value	Set the maximum value for the attribute.

RELATED TOPICS

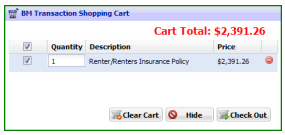
SHOPPING CARTS OVERVIEW

OVERVIEW

The shopping cart displays the total number of items in the cart and the total cost for those items. Users can add multiple products to a shopping cart before checking out. Users can selectively checkout items from the shopping cart. Upon checkout, all data is carried into the newly created commerce transaction, where the products added to the shopping cart show up as line items. Users can also add items to existing quote from the shopping cart, similar to the 'Add from catalog' functionality in commerce.

A shopping cart is available per commerce process. The fullaccessuser can enable the shopping cart for a commerce process in the Shopping Cart Editor. By default, this functionality is disabled.

Example of the User-Side Shopping Cart



Column	Description
Description	The description field is mapped to the Model Name for a Model and a Part Number for a part. The user cannot edit this field. Note, on an application supporting price books, the price book is displayed in parentheses next to the part number under the description column.
Price	The price field contains the Total Price for a configured Model. If the user adds a part and price books are not available, then the Price field contains the price associated with the part. If price books are available, the Price field contains the price of the part from the chosen price book. For a recommended item, the price field contains the recommended price. The end user cannot edit this field in the shopping cart.
Quantity	For recommended items and for items chosen in the Multiple Parts Buy Page, this field is populated by the recommended quantity/user specified quantity. The fullaccessuser can choose to make this field editable. However, the user cannot modify the quantity of mandatory recommended items. Also, changing the quantity of a configured model will change the quantities of mandatory items to keep the same ratio to the configured model but will not modify the quantities of non-mandatory recommended items.
Cart Total	The Cart Total provided in the Shopping Cart is a sum of item quantity multiplied by the unit price for each item. It does not allow advanced functions.

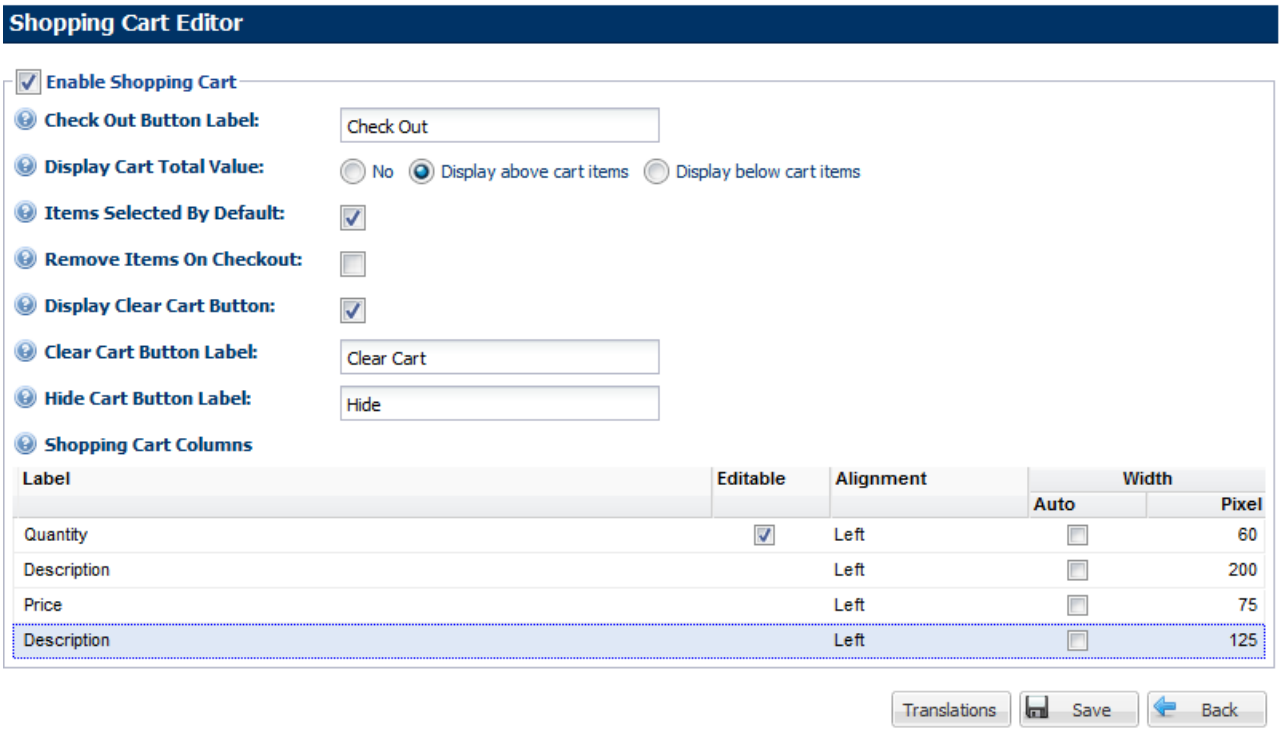
ADMINISTRATION

Enabling the Shopping Cart

1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Select a Process > Shopping Cart > List.
2. Verify that **Enable Shopping Cart** is selected.
3. Modify the Shopping Cart properties:

Property	Description
Check Out Button Label	Enter text that will display on the button users will click to perform the checkout action.
Display Cart Total	• No • Display Above Cart Item

Property	Description
	• Display Below Cart Item
Items Selected by Default	Select the Items Selected by Default box if you would like the items in the shopping cart to automatically be selected for check-out when they are added to the cart. If this is not selected, the user will have to manually select items for checkout.
Remove Items on Checkout	Select to have the selected items removed from the cart when they are added to a transaction.
Display Clear Cart Button	Selectthe Display Clear Cart Button if you would like that option. When users choose to clear the cart, all items will be removed from the cart.
Clear Cart Button Label	Enter the text that will be displayed on the Clear Cart button.
Hide Cart Button Label	Enter the text that will be displayed on the Hide Cart button.
Shopping Cart Columns	You can edit the text and the layout of the shopping cart. Columns can be dragged and dropped into the order you would like to have them appear. Under the Shopping Cart Columns header, there are four editable columns: Label, Editable, Alignment and Width. Under Label, you can click on the column label and edit its text.
Quantity	Select the checkbox under Editable if you would like the user to be able to edit the quantity of items within the shopping cart. In the Alignment column, click on a label for the appropriate shopping cart column. A drop-down menu will appear and you can choose Left, Right or Center for the alignment of each column.
Description	In the Alignment column, click on a label for the appropriate shopping cart column. A drop-down menu will appear and you can choose Left, Right or Center for the alignment of each column. You have two choices for setting the Width of each column: Auto and Pixel. If you click the checkbox under Auto, that column will auto-size.
Price	
Description	



4. Click **Save** to save your changes.
5. Click **Back** to return to the Processes List page.

Invoking the Shopping Cart

Commerce Invocation from the Shopping Cart is similar to invocation in other areas of the application. After enabling the Shopping Cart for a process, the fullaccessuser must set up the Shopping Cart invocation in the Process Invocation section. Invocation rules are evaluated on adding items to the cart and not on checking out. If the shopping cart for a process is disabled, the invocation buttons for it will also be disabled to make administration easier.

Shopping Cart and Navigation Menus

Also upon enabling a shopping cart within a commerce process and deploying the process, a navigation menu link is automatically created. It is labeled 'View <commerce process name> Shopping Cart'. The number of items in the shopping cart is displayed right next to the link for the end-user. The fullaccessuser can customize this link to be whatever they want. Also, like invocation, these view cart links disappear once the shopping cart is disabled for a process to make administration easier.

NOTES



The user cannot modify the quantity of mandatory recommended items. Changing the quantity of a configured model will change the quantites of mandatory items to keep the same ratio to the configured model but will not modify the quantities of non-mandatory items.

Shopping Cart is session based, therefore the user will lose all cart contents if he logs out.

It is NOT possible to remove the price book name from the parts description field within the Shopping Cart.



As of BigMachines 12, users can specify a quantity up to 9999999 in the shopping cart.

RELATED TOPICS

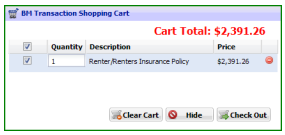
SHOPPING CARTS OVERVIEW

OVERVIEW

The shopping cart displays the total number of items in the cart and the total cost for those items. Users can add multiple products to a shopping cart before checking out. Users can selectively checkout items from the shopping cart. Upon checkout, all data is carried into the newly created commerce transaction, where the products added to the shopping cart show up as line items. Users can also add items to existing quote from the shopping cart, similar to the 'Add from catalog' functionality in commerce.

A shopping cart is available per commerce process. The fullaccessuser can enable the shopping cart for a commerce process in the Shopping Cart Editor. By default, this functionality is disabled.

Example of the User-Side Shopping Cart



Column	Description
Description	The description field is mapped to the Model Name for a Model and a Part Number for a part. The user cannot edit this field. Note, on an application supporting price books, the price book is displayed in parentheses next to the part number under the description column.
Price	The price field contains the Total Price for a configured Model. If the user adds a part and price books are not available, then the Price field contains the price associated with the part. If price books are available, the Price field contains the price of the part from the chosen price book. For a recommended item, the price field contains the recommended price. The end user cannot edit this field in the shopping cart.
Quantity	For recommended items and for items chosen in the Multiple Parts Buy Page, this field is populated by the recommended quantity/user specified quantity. The fullaccessuser can choose to make this field editable. However, the user cannot modify the quantity of mandatory recommended items. Also, changing the quantity of a configured model will change the quantities of mandatory items to keep the same ratio to the configured model but will not modify the quantities of non-mandatory recommended items.
Cart Total	The Cart Total provided in the Shopping Cart is a sum of item quantity multiplied by the unit price for each item. It does not allow advanced functions.

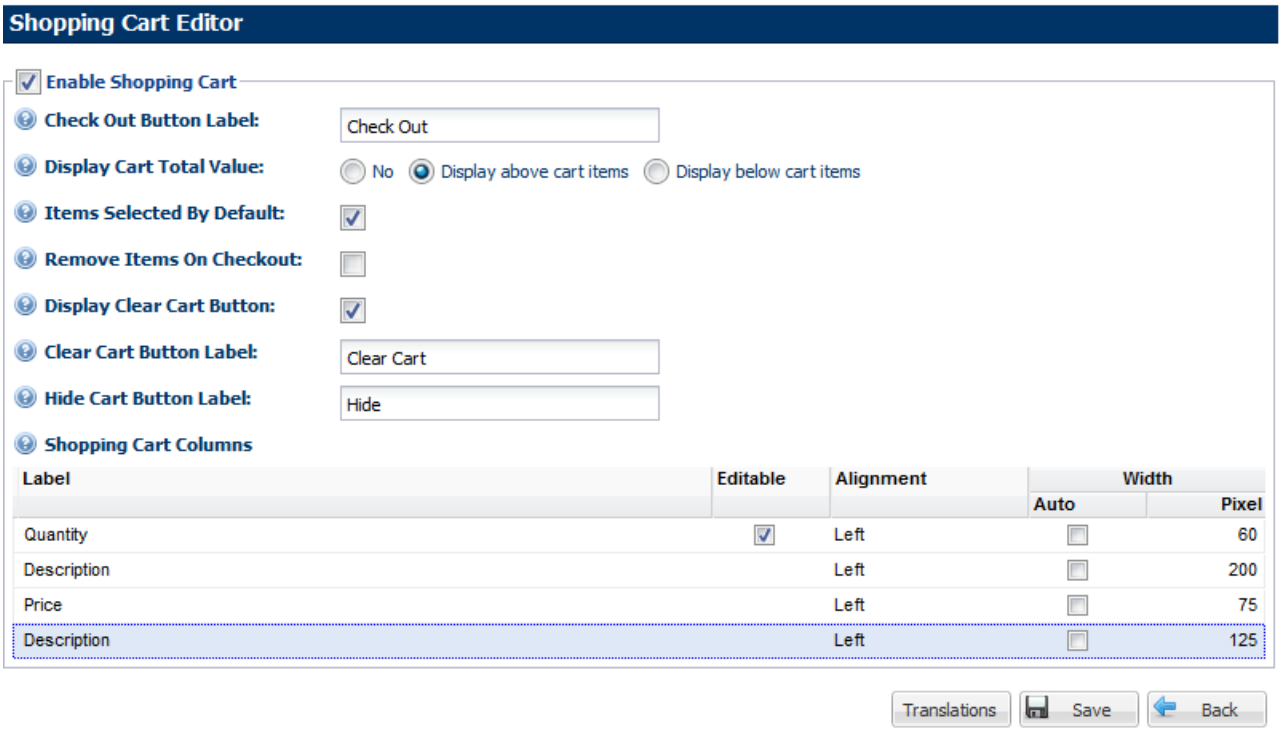
ADMINISTRATION

Enabling the Shopping Cart

1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Select a Process > Shopping Cart > List.
2. Verify that **Enable Shopping Cart** is selected.
3. Modify the Shopping Cart properties:

Property	Description
Check Out Button Label	Enter text that will display on the button users will click to perform the checkout action.
Display Cart Total	• No • Display Above Cart Item

Property	Description
	• Display Below Cart Item
Items Selected by Default	Select the Items Selected by Default box if you would like the items in the shopping cart to automatically be selected for check-out when they are added to the cart. If this is not selected, the user will have to manually select items for checkout.
Remove Items on Checkout	Select to have the selected items removed from the cart when they are added to a transaction.
Display Clear Cart Button	Selectthe Display Clear Cart Button if you would like that option. When users choose to clear the cart, all items will be removed from the cart.
Clear Cart Button Label	Enter the text that will be displayed on the Clear Cart button.
Hide Cart Button Label	Enter the text that will be displayed on the Hide Cart button.
Shopping Cart Columns	You can edit the text and the layout of the shopping cart. Columns can be dragged and dropped into the order you would like to have them appear. Under the Shopping Cart Columns header, there are four editable columns: Label, Editable, Alignment and Width. Under Label, you can click on the column label and edit its text.
Quantity	Select the checkbox under Editable if you would like the user to be able to edit the quantity of items within the shopping cart. In the Alignment column, click on a label for the appropriate shopping cart column. A drop-down menu will appear and you can choose Left, Right or Center for the alignment of each column.
Description	In the Alignment column, click on a label for the appropriate shopping cart column. A drop-down menu will appear and you can choose Left, Right or Center for the alignment of each column. You have two choices for setting the Width of each column: Auto and Pixel. If you click the checkbox under Auto, that column will auto-size.
Price	
Description	



4. Click **Save** to save your changes.
5. Click **Back** to return to the Processes List page.

Invoking the Shopping Cart

Commerce Invocation from the Shopping Cart is similar to invocation in other areas of the application. After enabling the Shopping Cart for a process, the fullaccessuser must set up the Shopping Cart invocation in the Process Invocation section. Invocation rules are evaluated on adding items to the cart and not on checking out. If the shopping cart for a process is disabled, the invocation buttons for it will also be disabled to make administration easier.

Shopping Cart and Navigation Menus

Also upon enabling a shopping cart within a commerce process and deploying the process, a navigation menu link is automatically created. It is labeled 'View <commerce process name> Shopping Cart'. The number of items in the shopping cart is displayed right next to the link for the end-user. The fullaccessuser can customize this link to be whatever they want. Also, like invocation, these view cart links disappear once the shopping cart is disabled for a process to make administration easier.

NOTES



The user cannot modify the quantity of mandatory recommended items. Changing the quantity of a configured model will change the quantites of mandatory items to keep the same ratio to the configured model but will not modify the quantities of non-mandatory items.

Shopping Cart is session based, therefore the user will lose all cart contents if he logs out.

It is NOT possible to remove the price book name from the parts description field within the Shopping Cart.



As of BigMachines 12, users can specify a quantity up to 9999999 in the shopping cart.

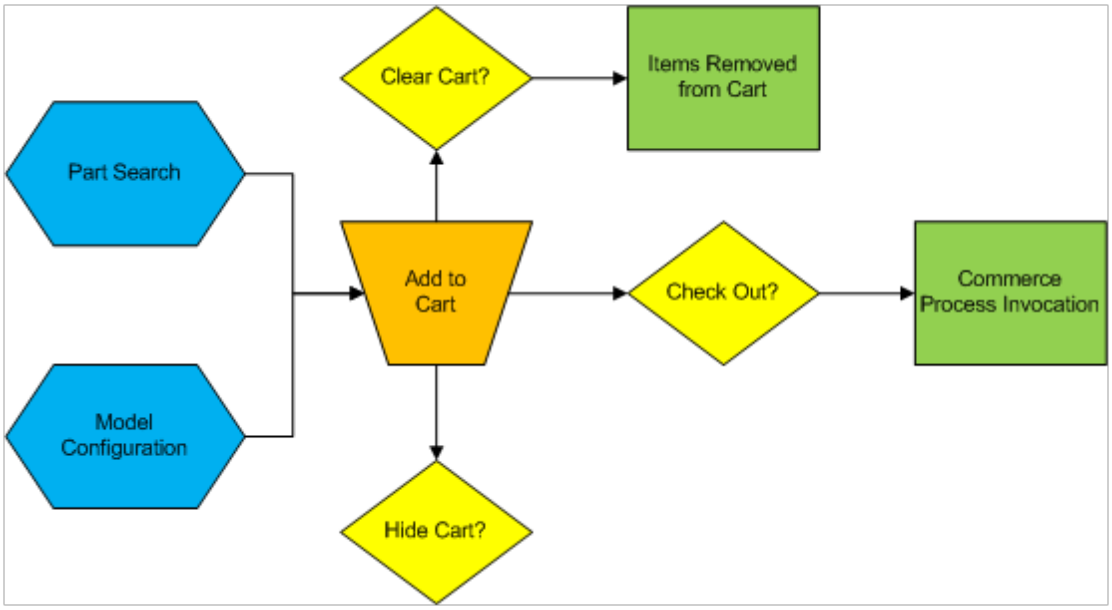
RELATED TOPICS

SHOPPING CARTS - USER SIDE FUNCTIONALITY

OVERVIEW

The Shopping Cart is an e-commerce feature designed to allow users to search for items or configure items and add them to a cart and stay on the same page. This allows users to create a quote with multiple configured items. The shopping cart displays all of your parts and configured items, as well as mandatory and recommended items, descriptions, product prices and a cart total. Users can clear their cart and start over or hide the cart from view until they are satisfied with their selections. Users can also modify quantities of non-mandatory items before adding them to a transaction. When users click Check Out, a commerce process will be invoked.

Standard User-Side Shopping Cart Workflow



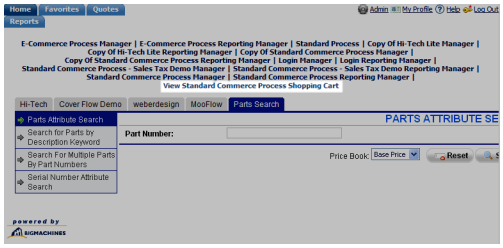
User-Side UI

All user-side pages will be affected if the shopping cart functionality is turned on by the fullaccessuser. Enabling the Shopping Cart will allow the user to create a quote from their cart, regardless of where they are in the application. All user-side pages will have the "View Cart" navigation link available and is available on ALL configuration flows.

Example of the Shopping Cart



Example of the View Cart Navigation Link

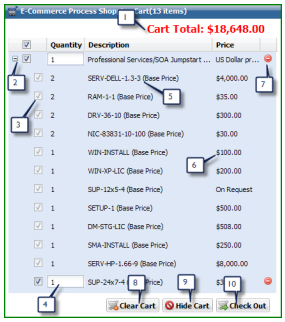


View Cart Panel

The shopping cart view panel displays the current cart items. This panel is a floating, inline panel which is shown whenever a users clicks on the Add to Cart or the View Cart navigation link.

Overview of Shopping Cart

#	What Is It?	What Does It Do?
1	**Cart Total**	Displays the total value of the items in the cart. This is a simple sum and cannot be edited.
2	**Expand/Collapse Icon**	Expand or Collapse to show/hide all recommended and mandatory items.
3	**Checkbox to Select Items**	Select items for check-out. Un-selected itmes will not be processed (added to the transaction) when the commerce process is invoked.
4	**Quantity**	Displays the quantity of each item and ultimately will set the quantity of each line item in commerce. The quantity will always default to 1, unless otherwise specified by recommended items. Recommended item quantities can be edited by the user, but mandatory item quantities are read-only.
5	**Description**	Displays the model name and/or part number for the item.
6	**Price**	Price of the part(s) or configured model.
7	**Delete Icon**	Removes selected items from the cart.
8	**Clear Cart**	Removes ALL items from the cart.
9	**Hide Cart**	Select to hide the View Cart panel. It can be viewed again by clicking on the View Cart Navigation link or when a user adds additional items to the cart.
10	**Checkout**	Click to run a commerce process invocation rule that will create and add line items to a transaction. There must be at least one item in the cart.



NOTES



Cart items only exist for the duration of the browser session.

RELATED TOPICS

SHOPPING CART - FAQ'S

I've enabled the Shopping Cart and deployed my process. Why can't I see Add to Cart or View Cart links in the user side?

Add to Cart buttons are process invocations, just like for commerce processes themselves. The admin must create these under Process Invocation. The only special step required is to set the invocation type to "Shopping Cart" upon creation.

View Cart links are created in Navigation Menus where they can be customized similar to quote manager links. A standard navigation menu link is created when shopping cart is enabled; the navigation menus must be manually deployed for this change to take effect.

If this occurs when adding items to an existing quote, verify that "Hide Commerce Invocation Buttons" on "Add from Catalog" is set to No. This setting can be found in Global System Settings and Utilities > Commerce.

I can see Add to Cart buttons in some areas of configuration but not others.

Verify process invocation is set up for all desired areas and correct user permissions are given.

After logging out and back in, the contents of my shopping cart are gone.

Shopping carts are saved per session. Logging out ends that session. This is correct behavior.

Can I use Shopping Cart when adding items to an existing quote?

Yes. When used in this case, the cart has special buttons for "Add to Quote" and "Return to Quote"

RELATED TOPICS

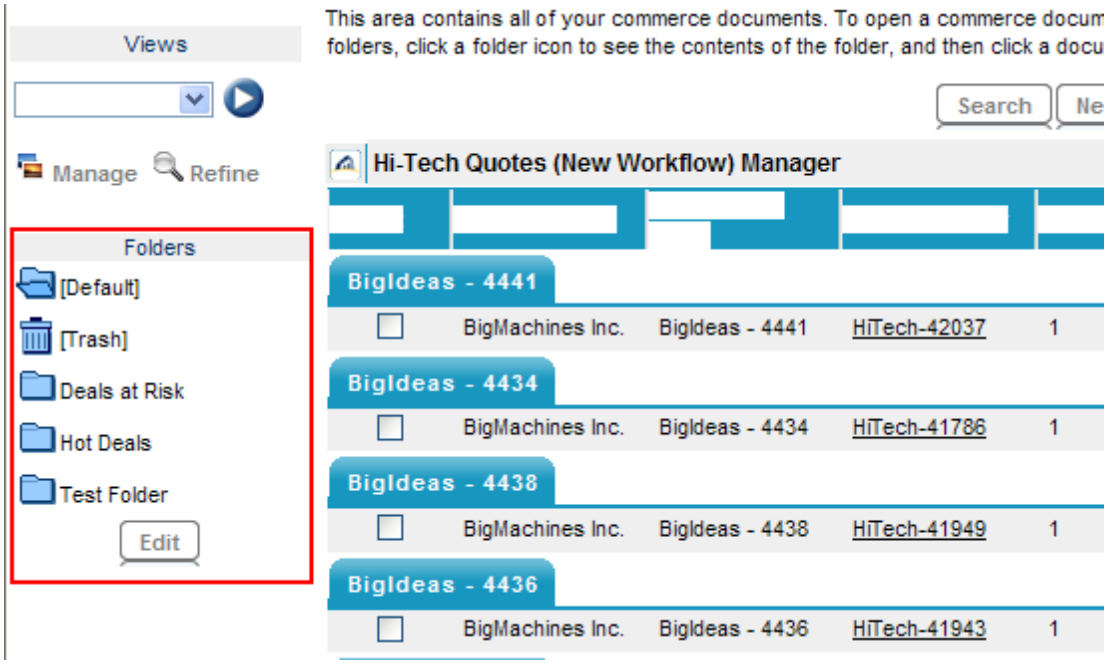
TRANSACTION MANAGER

OVERVIEW

A Transaction Manager is a list of transactions or quotes that you can access. This is most frequently where approvers will go to approve quotes and where you can download your quotes or transactions.

BigMachines automatically creates a Transaction Manager for each commerce process. **For example:** "[Commerce Process Name] Manager". Individual users may have access to different information within the transaction manager based on their participant profile and the step the transaction is in.

You can re-size the columns in the Transaction Manager. You can also sort the columns by clicking on the specific column header. This will sort the data and display the sorting order next to the column header (ascending or descending).



ADMINISTRATION

Action Bar

There is an action bar at the top and bottom of each Transaction Manager. These are Process Actions, which trigger events within a transaction manager.

Views

BigMachines can search transaction data based on commerce main document and line item attribute data as long as the attributes have been mapped in the data columns. Search results will be returned based on user permissions.A View Manager will be made available for every commerce process in the system. The user can save searches and use them to creates Reports. Saved searches are known as Views.

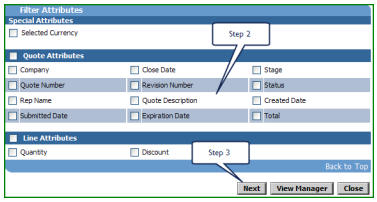
The maximum number of saved views is 20.

Commerce attributes mapped in the data columns and process manager can be used as filter inputs to the search. The user can specify criteria for the attributes so that the search returns only those records that satisfy the criteria. You can also pass in blank values as criteria for the filter attributes. In that case, all records are returned.

If no Data Column Mapping are available, then no Process Manager Columns can be mapped. In that case, the Transaction Manager will show just the ID# of the quote created.

Creating Views

1. Navigate to the Transaction Manager > Views > Manage > Add.
2. Select the **Filter(s)** you'd like to apply to the search, based on Special, Quote and Line Attributes.
3. Click **Next** to proceed and select the columns you'll use to organize the view.



4. Select the **Column(s)** that you'll use to organize the view, based on Special and Quote Attributes.
5. Click **Next** to launch the Search Editor page.
6. Specify any **specific criteria for the filter attributes** selected in Step 2.
7. Name the Search if you plan on referring to it again. IMPORTANT: If you select View Manager, you will be returned to the Transaction Manager without saving your changes.
8. Click **Search**. You'll see the results returned based on your filter criteria and display columns selected.

If you need to create a view (or perform a search) on transactions, or if you have already created a view, you'll be able to do so from the View Manager

Folders

You have the ability to organize transactions by moving them to different folders. This can be done in two ways:

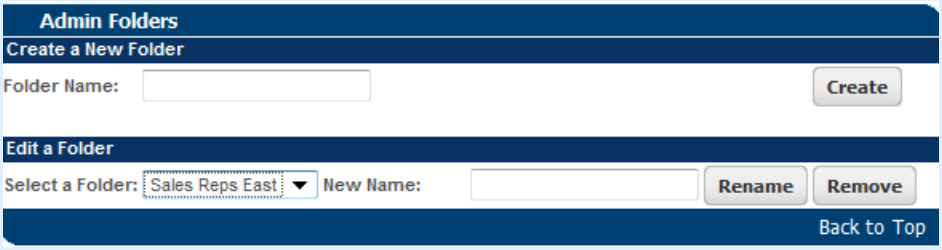
1. Select one or more transactions, then click then select the appropriate folder from the Move drop-down menu.
2. Select multiple transactions and then drag them from one folder to another. As you drag the row(s), a red icon will appear to show you where you cannot place them and a green icon will appear if you can drop them.

Creating Folders

1. Navigate from BigMachines to the Transaction Manager.
2. Click **Edit** under the Folders header in the left sidebar.
3. Enter a **folder name**.
4. Click the **Create** button.



Rename a folder by selecting it from the drop-down menu and entering a new name. You can remove a folder by selecting it from the drop-down menu and clicking Remove.



5. Click Back to return to the Transaction Manager.

NOTES



Individual users will want to use View and Search functionality to organize and manage their transactions.



Process Manager Columns are displayed in the Transaction Manager in the order that they are maintained in the Process Manager Columns page.

RELATED TOPICS

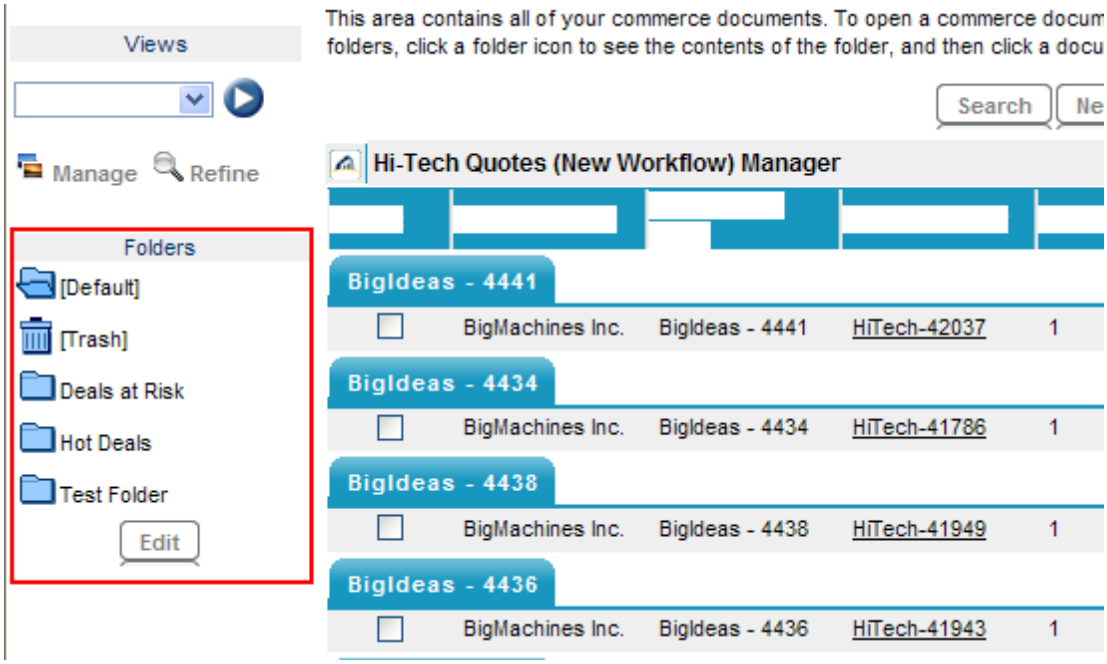
TRANSACTION MANAGER

OVERVIEW

A Transaction Manager is a list of transactions or quotes that you can access. This is most frequently where approvers will go to approve quotes and where you can download your quotes or transactions.

BigMachines automatically creates a Transaction Manager for each commerce process. **For example:** "[Commerce Process Name] Manager". Individual users may have access to different information within the transaction manager based on their participant profile and the step the transaction is in.

You can re-size the columns in the Transaction Manager. You can also sort the columns by clicking on the specific column header. This will sort the data and display the sorting order next to the column header (ascending or descending).



ADMINISTRATION

Action Bar

There is an action bar at the top and bottom of each Transaction Manager. These are Process Actions, which trigger events within a transaction manager.

Views

BigMachines can search transaction data based on commerce main document and line item attribute data as long as the attributes have been mapped in the data columns. Search results will be returned based on user permissions.A View Manager will be made available for every commerce process in the system. The user can save searches and use them to creates Reports. Saved searches are known as Views.

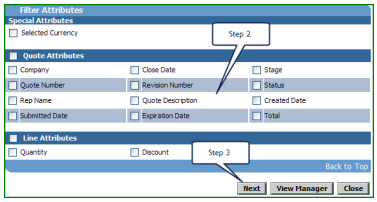
The maximum number of saved views is 20.

Commerce attributes mapped in the data columns and process manager can be used as filter inputs to the search. The user can specify criteria for the attributes so that the search returns only those records that satisfy the criteria. You can also pass in blank values as criteria for the filter attributes. In that case, all records are returned.

If no Data Column Mapping are available, then no Process Manager Columns can be mapped. In that case, the Transaction Manager will show just the ID# of the quote created.

Creating Views

1. Navigate to the Transaction Manager > Views > Manage > Add.
2. Select the **Filter(s)** you'd like to apply to the search, based on Special, Quote and Line Attributes.
3. Click **Next** to proceed and select the columns you'll use to organize the view.



4. Select the **Column(s)** that you'll use to organize the view, based on Special and Quote Attributes.
5. Click **Next** to launch the Search Editor page.
6. Specify any **specific criteria for the filter attributes** selected in Step 2.
7. Name the Search if you plan on referring to it again. IMPORTANT: If you select View Manager, you will be returned to the Transaction Manager without saving your changes.
8. Click **Search**. You'll see the results returned based on your filter criteria and display columns selected.

If you need to create a view (or perform a search) on transactions, or if you have already created a view, you'll be able to do so from the View Manager

Folders

You have the ability to organize transactions by moving them to different folders. This can be done in two ways:

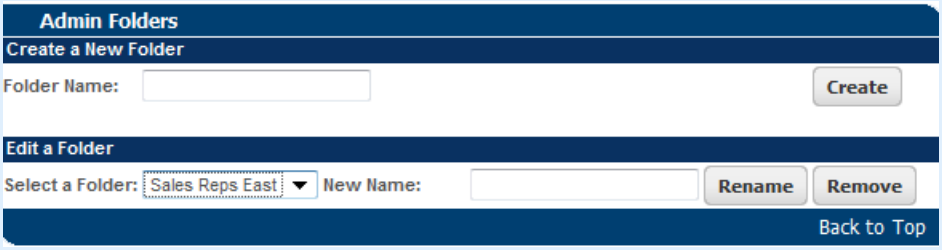
1. Select one or more transactions, then click then select the appropriate folder from the Move drop-down menu.
2. Select multiple transactions and then drag them from one folder to another. As you drag the row(s), a red icon will appear to show you where you cannot place them and a green icon will appear if you can drop them.

Creating Folders

1. Navigate from BigMachines to the Transaction Manager.
2. Click **Edit** under the Folders header in the left sidebar.
3. Enter a **folder name**.
4. Click the **Create** button.



Rename a folder by selecting it from the drop-down menu and entering a new name. You can remove a folder by selecting it from the drop-down menu and clicking Remove.



5. Click Back to return to the Transaction Manager.

NOTES



Individual users will want to use View and Search functionality to organize and manage their transactions.



Process Manager Columns are displayed in the Transaction Manager in the order that they are maintained in the Process Manager Columns page.

RELATED TOPICS

PROCESS MANAGER

OVERVIEW

Process Manager Columns specify the main document and/or line item attribute data that is displayed in the columns available in the Transaction Manager page.

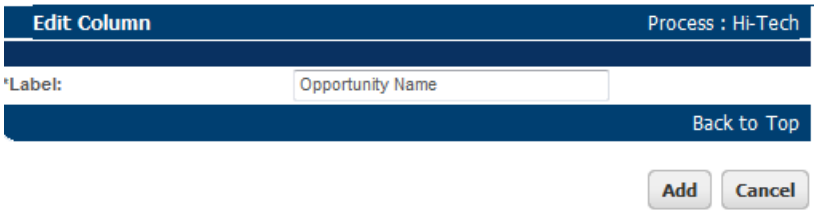
The following must be in place for columns to be available for display in the Transaction Manager:

1. The main document or line item attribute must be mapped to the relevant document in the data columns.
2. The Process Manager Column is mapped to the data column mapping of the main document or line item attribute.

ADMINISTRATION

Adding a Process Manager Column

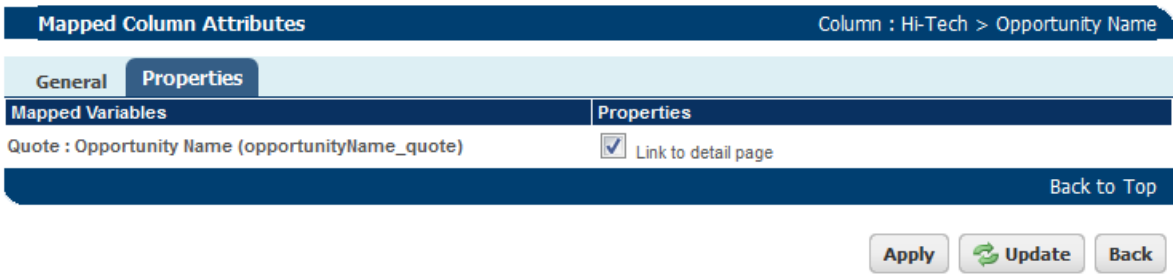
1. Navigate from the Admin Home Page > Commerce and Documents > Select a Process > Process Manager > List
2. Click **Add** on the Process Manager Columns page.
3. Enter the **column name** in the Label field. *This name appears at the top of the column on the Transaction Manager page.*



4. Click **Add**. *The new column appears on the Process Manager Columns page.*
5. Click the **Column Mapping link** next to the attribute you would like to map to a data column.



6. Choose the **data column(s)** from the Available Attributes Menu and **click (>)** arrow to move them to the Column Contents field.
7. Click the Up and Down arrows to move the data columns up or down.
8. Click **Apply** to save changes and refresh the page.
9. Click the **Properties** tab.
10. Select the *Link to detail page* box to create a link to the transaction from the data in the column.



11. Click **Update** to save the changes and return to the Process Manager Columns page.

Reordering Data Columns

1. Specify the **order number** of the mapped data column on the Process Manager Columns page.
2. Click **Apply.**

Translations

1. Click the **Translations button** on the Mapped Data Columns page.
2. Enter the translation for the **Label** for each supported language.
3. Click **Save and Close** to return to the Mapped Data Columns page.

NOTES

New Columns are not visible until the process is deployed (or redeployed). Before deploying a new commerce process, manage all components of the process.



Column mappings, renamed columns or other changes are not visible until the process is deployed (or redeployed).

After creating columns, you can order how they appear on the Transaction Manager page.



To map or reorder columns in a deployed commerce process, contact an implementation engineer. This functionality is hidden from administration after the commerce process is deployed.

Use caution when deleting a column because it is not possible to recover the deleted data. Upon deletion, all associated column mapping are also deleted. To delete columns from a Transaction Manager in a deployed commerce process, contact an Implementation Engineer. This functionality is hidden from administration after the commerce process is deployed.

If no Data Column Mapping are available, then no Process Manager Columns can be mapped. In that case, the Transaction Manager will show just the ID# of the quote created.

RELATED TOPICS

PROCESS ACTIONS

OVERVIEW

Process actions trigger events within a Transaction Manager to occur. The BigMachines system provides a standard set of customizable process actions for each commerce process.

Each action can be customized with different options, depending on the action's functionality:

- For all actions, you can define action labels and create certain rules depending on the action.
- For print and email actions, you can design custom forms using XSL Stylesheets.
- For the copy action, you can define initialization options for copied transactions and generate one or more copy options using sequences.

Standard Process Actions

Process Action Name	Variable Name	Description

Copy	_action_process_resubmit	The Copy action enables users to duplicate commerce transactions. Multiple copy options can be offered to users, permitting people to copy up until a certain point in a transaction.
		• **Initialization:** Specify how to populate attribute values in documents copied from other transactions. The Initialization tab contains tabs for each document that can be copied in the process. On each tab, you can set how you want the system to populate document attributes during a copy process.
		  • **Use Specified Value :** This option enables you to hard-code an attribute's value using the available text field.
		  • **Use Function:** This option can be used to define an advanced function for setting an attribute's value.
		  • **Revert to Default:** When the Revert to Default option is enabled, the system uses the default value set for the attribute itself. This option is useful for picking up the system's current date and time, one of the many uses you can choose this option for.
		  • **Copy from Original Doc:** This option means the system will set the attribute's value using the value set in the original document. When this option is used, mapping settings associated with the create document action are not used.
		• **Process Views** The Email action can also be displayed based on the process view tab settings or its visibility settings. The Process View Tab Settings will control the visibility of the Email action based on every step of the transaction. The visibility setting of this action will allow it to be always hidden or always displayed.

Email	_action_process_email_print	The Email action enables users to access email-friendly document formatting. • **XSL Views:** This action can be associated with one or more XSL views. If multiple XSL views are made available with the action, users can choose the email form they want to use. In order to link XSL Stylesheets with these actions, the XSL files must be developed offline and then uploaded into the XSL administration area. • **Process Views** The Email action can also be displayed based on the process view

Process Action Name	Variable Name	Description
		tab settings or its visibility settings. The Process View Tab Settings will control the visibility of the Email action based on every step of the transaction. The visibility setting of this action will allow it to be always hidden or always displayed.
Export File Attachments	_action_export_file_attachments	Exports file attachments via Web Services.
Print	_action_import_file_attachments	The Print action enables users to access email-friendly document formatting. • **XSL Views**: This action can be associated with one or more XSL views. If multiple XSL views are made available with the action, users can choose the email form they want to use. • **Process Views:** The Print action can also be displayed based on the process view tab settings or its visibility settings. The Process View Tab Settings will control the visibility of the action based on every step of the transaction. The visibility setting of this action will allow it to be always hidden or always displayed.

Forward	_action_process_forward	The Forward action enables users to forward commerce documents to users or groups (configured in the Users Administration and Groups Administration areas).
		• **Modify:** The Forward action provides a modify tab, which can be used to initialize attribute value settings for every document in a process, at the time the transaction is forwarded to users or groups.
		  • **Define Function:** This option can be used to fine an advanced function for setting the attribute's value.
		  • **Use Formula:** choose this option to modify the action based on a formula created using Formula Management.
		  • **Use Specified Value :** This option enables you to hard-code an attribute's value using the available text field.
		  • **Leave Value Unchanged.**
		  • **Revert to Default:** When the Revert to Default option is enabled, the system uses the default value set for the attribute itself. This option is useful for picking up the system's current date and time, one of the many uses you can choose this option for.
		  • **Save from Form**: Saves values that were entered by the user.
		• Integration
		• **Process Views:** The Forward action can also be displayed based on the Process View Tab settings or its visibility settings. The Process View Tab Settings will allow you to control the visibility of the Forward action on every step of the transaction. The visibility setting of this action will allow it to be always hidden or always displayed.

Move	_action_process_move	This action enables users to move commerce documents into different folders. A move action appears as a button next to a drop-down list box containing folder names.
Import File Attachments	action_import_file_attachments	Imports file attachments via Web Services.
New Transaction	_action_new_transaction	The New Transaction action enables users to create a new commerce transaction without first having to add a line item. This action appears as a button at the bottom of the Commerce Manager.
Open Quote	_open_quote_process	This action enables users to open documents listed on the Commerce Manager page. The BigMachines system dynamically generates an open action for every document added to a commerce process. You can choose the document attribute you want users to click to trigger this action. • **Modify:** Use to initialize attribute value settings for every document in a process, at the time a process is opened.

Process Action Name	Variable Name	Description
		• **Define Function:** This option can be used to fine an advanced function for setting the attribute's value. • **Use Formula:** choose this option to modify the action based on a formula created using Formula Management. • **Use Specified Value :** This option enables you to hard-code an attribute's value using the available text field. • **Leave Value Unchanged.** • **Revert to Default:** When the Revert to Default option is enabled, the system uses the default value set for the attribute itself. This option is useful for picking up the system's current date and time, one of the many uses you can choose this option for. • **Save from Form**: Saves values that were entered by the user. • Integration • **Process Views:** This action can also be displayed based on the Process View Tab settings. The Process View Tab Settings will allow you to control the user permissions to view the document on every step of the transaction.

ADMINISTRATION

Setting Access Rights for the New Transaction action

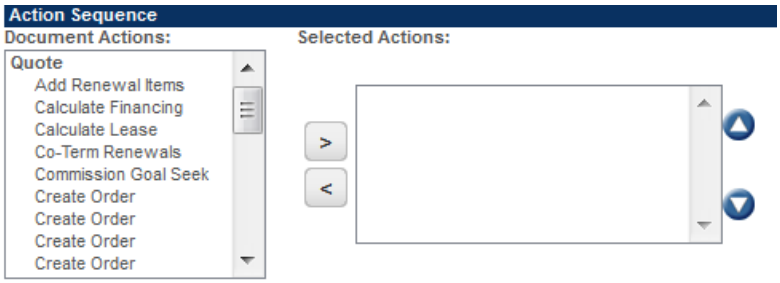
1. Navigate from the Admin Home Page > Commerce and Documents > Process Defintion > Process Actions.
2. Click the **Access Rights** link next to the New Transaction Action.
3. Choose a company type from the **Company Type** drop-down list who you want to be able to perform the Copy action.
4. Choose a user type from the **User Types** drop-down list who you want to be able to perform the Copy action.
5. Click the greater than **(>)** sign button for moving the choose Company Type:User Types set to the Access Rights list box.
6. Remove a Company Type:User Types set from the Access Rights list box by choosing the Company Type:User Types set and clicking the lesser than (<) sign button.
7. Click **Apply** to save the changes made. Click Update to display the modified data on the page for customizing process actions. Click back to return without saving changes.

Managing Sequence Lists for the Copy Action

The Copy process lets you define how much of a transaction they want copied when the action is performed. You can offer different copying options to different user types. When multiple copy options are present, users can choose how much of a transaction they want copied. This flexibility is possible through the use of an administration component called the Sequence List.

A sequence informs the system which document actions to perform while copying a transaction and who should have access to the copy option. Users only see the copy options they have access rights to.When more than one sequence exists for a user type, a dialog lets the user to choose a copy option. When no sequence exists, the system performs the default copy behavior which is to copy the first step of the first document in a transaction.

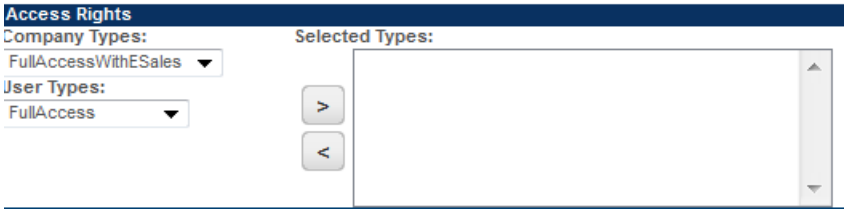
1. Navigate to the Process Action page.
2. Click **Sequence List**, next to Copy Action.
3. Enter a **Name and Description**.
4. Choose the **document actions** you want the system to perform for the Copy action. To do so, under the Action Sequence section:
    - Choose a quote or line-level action name from the list available under Document Actions list box
    - Click the greater than (>) sign button for moving the selected action to the Selected Actions list box.
    - Order the Document Actions in the way they would be performed in a transaction workflow in the Selected Actions list box:
        - Choose the document action in the Selected Actions list box and click to move the actions up or down.
    - Remove document actions from the Selected Actions list box by choosing the document action name and clicking the lesser than (<) sign button.



5. Choose the **Company Types and User Types** who you want to be able to perform the Copy action, from their respective drop-down menus.
6. Click the greater than (>) sign button for moving the chosen Company Types:User Types set to the Selected Types list box.



Remove by Company Types:User Types set and clicking the lesser than (<) sign button.



7. Click **Add**. This adds the copy sequence and displays it on the page for managing copy sequences.

8. Click **Back** to go back to the page for managing copy sequences.



The last document action performed by the system for a copy action should leave the document in a step where data can be saved. If the copy process takes the transaction through to the end, without letting the user perform a save (modify) type document action, the document will not be saved on the system.

NOTES



When configuring the system to copy two documents in a transaction, the "Create Document" type action must be included in the action sequence so the system can generate the second document.



Edits are not visible until the process is deployed (or redeployed). Before deploying a new commerce process, administrate all of its components.

The sequence name and description should enable users to distinguish one type of copy action from another. If there are multiple copy options available to a user, the sequence name and description distinguish one copy option from another in a dialog box whenever they click the copy process action.



Only full access users can manage process actions.

RELATED TOPICS

DATA COLUMNS

OVERVIEW

Data Columns are used to map attribute to document. Attributes can only be mapped to the document under which they are created. Data Column Mappings cannot have duplicates; an attribute can be mapped to only one document.

Data Columns are used to create Process Manager Column mappings. They are also useful in Reporting and Search because both these functions can only be performed on data of attributes that have been mapped in the data columns.

Each Commerce Process can have **251** Data Column mappings.

- **101** Data Columns are available for main-document level Attributes (100 Data Columns and 1 Tab Column, located on the Process Edit page).
- **150** Data Columns are available for sub-document level Attributes. *These Data Columns can be mapped to sub-document Attributes or Configuration Attributes.*



Mapped Data Columns — Process : Hi-Tech

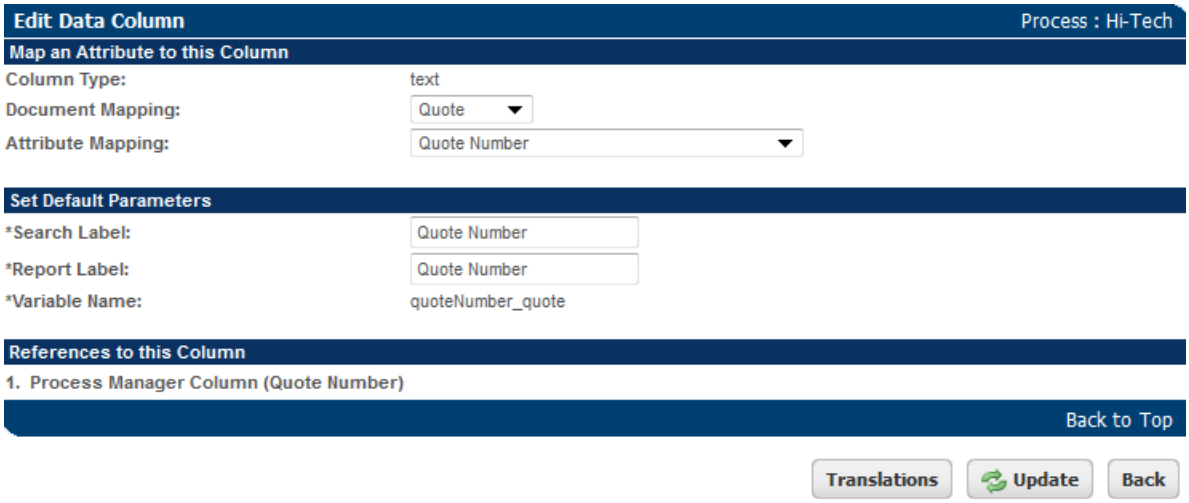
Select	Order	Attribute Type	Attribute	Search Label	Report Label	Restrict Access	Date Last Modified
	1	Commerce Attribute	Quote : Quote Number (quoteNumber_quote)	Quote Number	Quote Number		11/01/2011 11:39 AM
	2	Commerce Attribute	Quote : Company Name (_billTo_company_name)	Company	Company		11/01/2011 11:39 AM
	3	Commerce Attribute	Quote : Date of Approval 1 (dateOfApproval1_quote)	Date Of Approval 1	Date Of Approval 1		11/01/2011 11:39 AM
	4	Commerce Attribute	Quote : Opportunity Name (opportunityName_quote)	Opportunity Name	Opportunity Name		11/01/2011 11:39 AM
	5	Commerce Attribute	Quote : Revision Number (revisionNumber_quote)	Revision Number	Revision Number		11/01/2011 11:39 AM

ADMINISTRATION

Adding a Data Column

1. Navigate from the Admin Home Page > Commerce and Document > Process Definition > *Select a Process* > Data Columns > List.
2. Click **Add**on the Mapping Data Columns page.
3. Choose the **Document(quote or line)** to map an attribute to.
4. Choose the **Attribute** to map to, based on the document selected in the previous step.
5. Set **Default Parameters:**
    - **Search Label:**Enter the text label that should appear in any Search results.
    - **Report Label:** Enter the text label that should appear on Reports, if enabled.
    - **Variable Name**: Enter the variable name for the column. This cannot be changed.

If you return to edit at a later date, you'll see a list of where this data column is referenced. This includes Process Manager columns.



6. Click **Add** to save your changes or click Cancel to return to the Mapping Data Columns page without saving your changes.

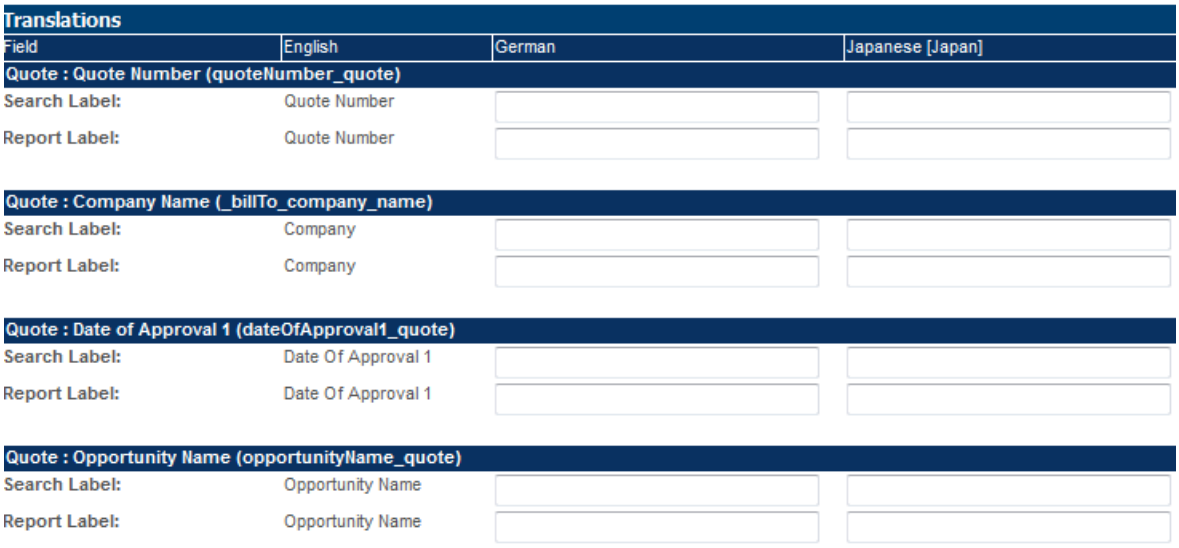
If you have made changes to Data Columns in a process that has already been deployed, make sure to go to the Deployment Center to repopulate column data.

Reordering Data Columns

1. Specify the **order number** of the mapped data column on the Mapped Data Columns page.
2. Click **Apply.**

Translations

1. Click the **Translations button** on the Mapped Data Columns page.
2. Enter the translation for the **Search Label** and/or **Report Label** for each supported language.

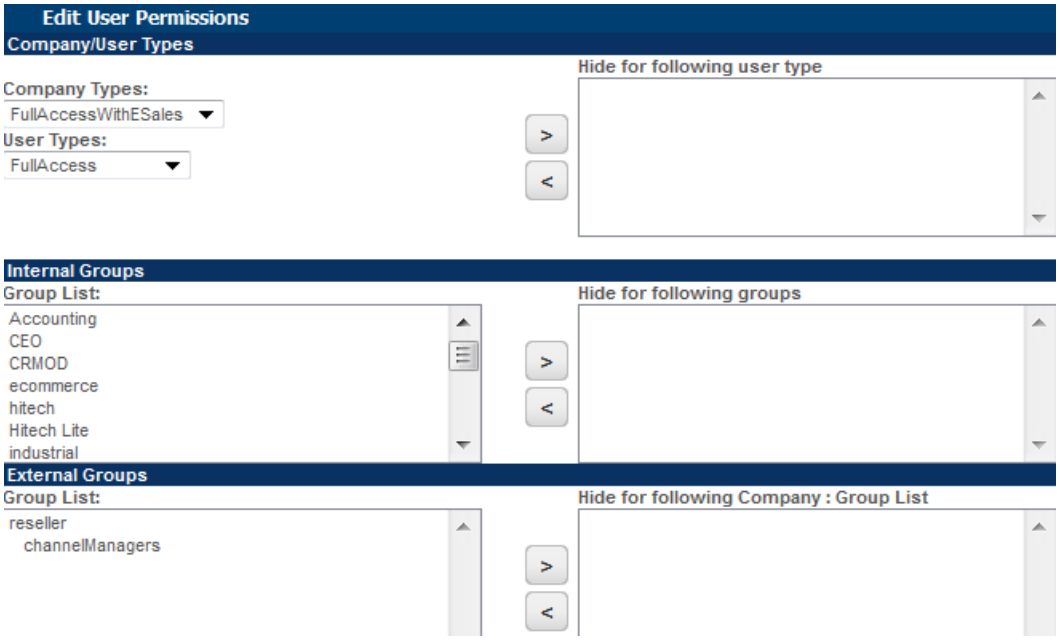


3. Click **Save and Close** to return to the Mapped Data Columns page.

Restricting Access for Data Columns

Data columns can be hidden from certain users or groups in Reporting, Search and the Process Manager. The fullaccessuser might choose to set up permissions on data columns to ensure that sensitive information mapped in the data columns is not visible to all users in Reporting, Search and/or Process Manager.

1. Click on the **Restrict Access Icon** for the selected **data column**. *A dialog will appear.*
2. Choose a company type from the Company Type drop-down list who you want to be able to perform the Copy action.
3. Choose a user type from the User Types drop-down list who you want to be able to perform the Copy action.
4. Click the greater than (>) sign button for moving the selected Company Type:User Types set to the *Hide for following user type* box.
5. Remove a Company Type:User Types set from the Access Rights list box by selecting the Company Type:User Types set and clicking the lesser than (<) sign button.
6. From the list(s) of internal/external groups, select the group(s) for whom you want to restrict access.
7. Move the user/group to the *Hide for Following* list(s).



8. Click **Save** to save changes and remain on the page. Click Save and Close to save changes and return to the Mapping Data Columns page. Click Close to return without saving changes.

To remove restricted access, simply select a user/group and click the button.

NOTES

Repopulate Column Data can be found in the Deployment Center. This event ensures that any changes made in the Data Columns on a deployed process are available on the commerce side. This event is basically applicable for the transaction manager data

Note on Attributes selected for Tab Label in the Process Manager: Tab Labels are always visible to every user even if this attribute is mapped to a data column that is hidden from the logged in user.

Note on Restricted Data Coluns as Condition Filters in Reporting/Search: Report/Search results are only run on the condition filter attributes that the user has access to in the data columns. The fullaccess user must be aware that shared reports/searches might yield erroneous results for certain users if these



users have been restricted from data columns that are used as condition filters in the shared reports/searches.



Data columns cannot be deleted after they are created.

RELATED TOPICS

MY APPROVAL SEARCH

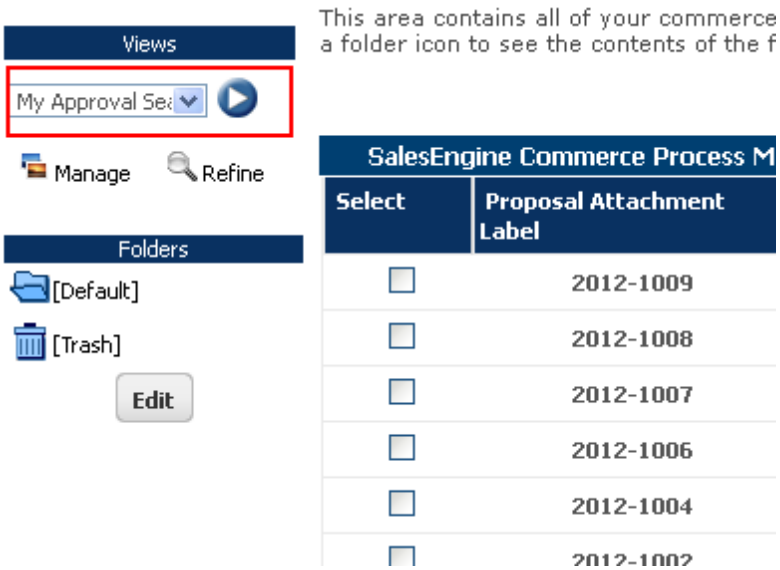
OVERVIEW

BigMachines allows users to search for pending approvals through the View Manager. Users have pending approvals waiting in a queue when there are quotes they need to approve. This makes it easy to enter each individual quote to approve or reject them.

ADMINISTRATION

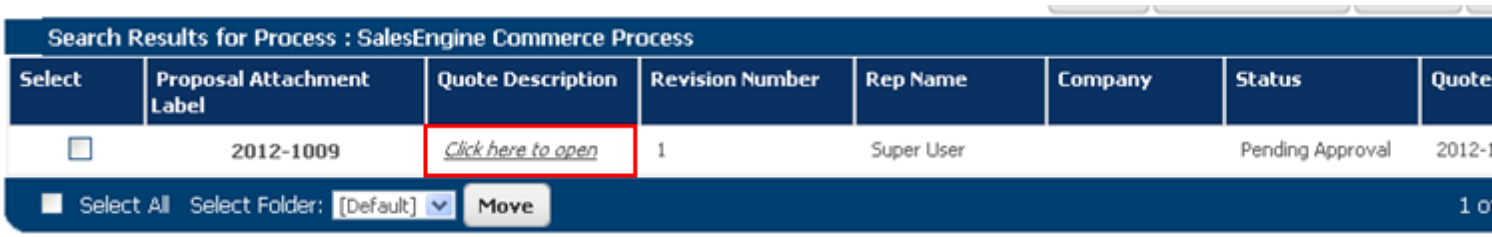
Viewing Pending Approvals

1. Navigate to the Transaction Manager from the header or side navigation menu.
2. Select **My Approval Search** from the Views drop-down menu and click the **Blue Arrow** to perform the search.

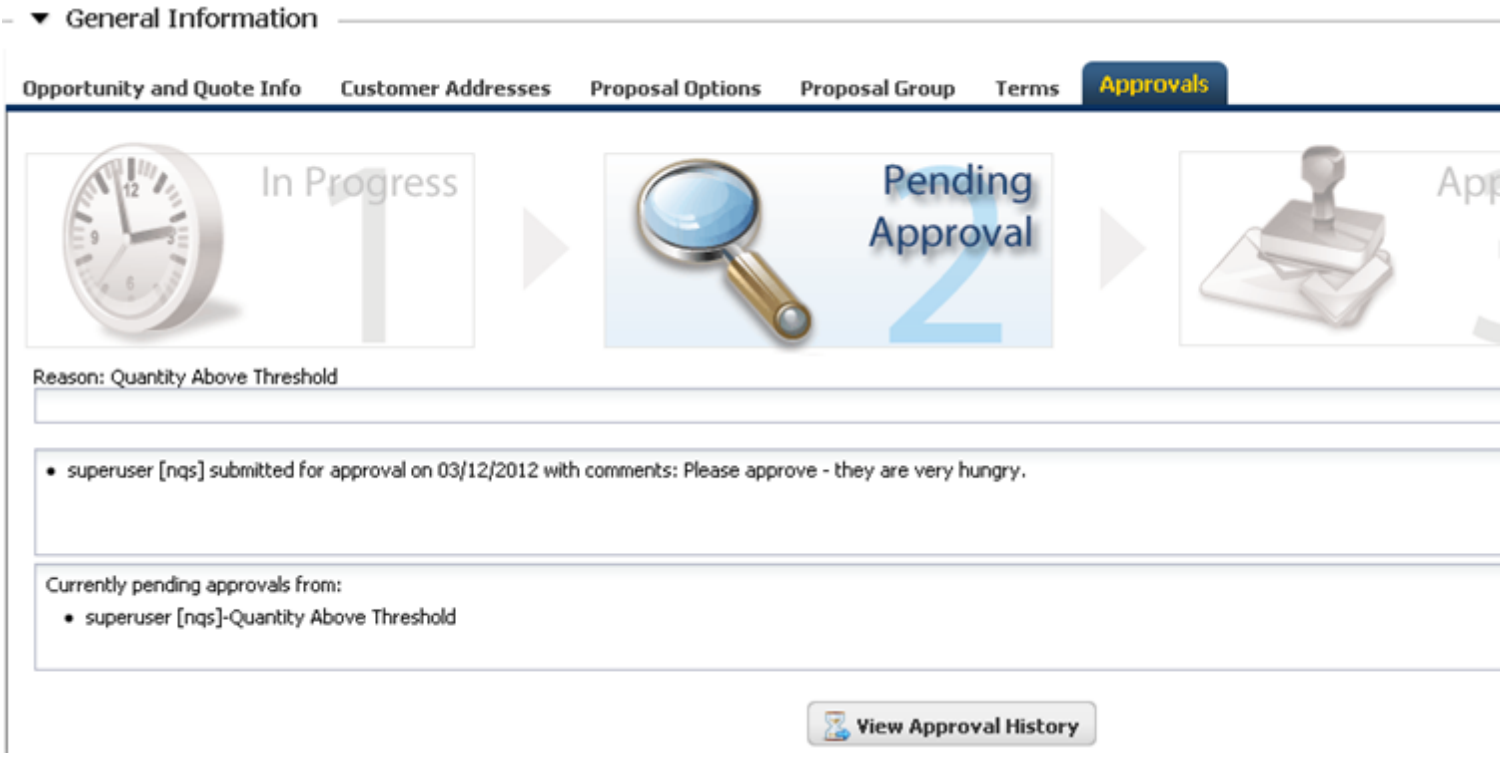


You'll notice a list of your pending approvals in the transaction manager search results page.

3. Click the ***Click here to open*** link under the Quote Description column.This will take you directly to the quote you need to approve.



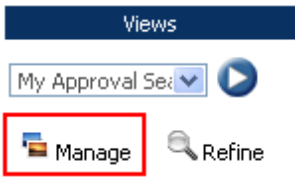
4. Click on the **Approvals tab** on the main-document and either approve or reject the quote.



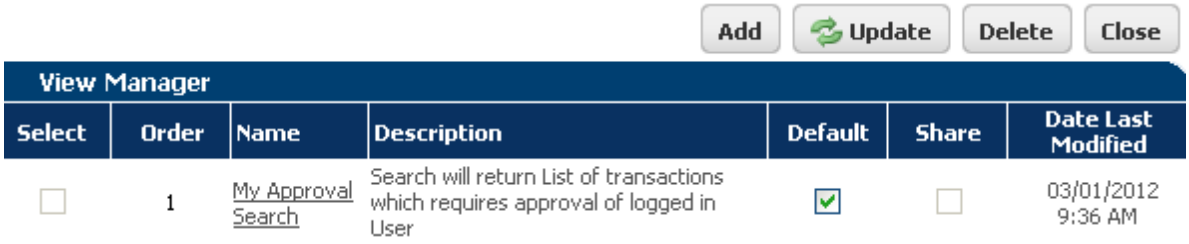
Managing My Approval Search View

You can set My Approval Search as the default search view on the transaction manager. In doing so, when the user clicks on the transaction manager, they will always see their pending approvals first.

1. Navigate to the **Transaction Manager** from the header or side navigation menu.
2. Click **Manage** underneath the Views header.



3. Select the **Default checkbox** to set My Approval Search as the default view.



4. Click on My Approval Search under the Name column to change the name of the search.

RELATED TOPICS

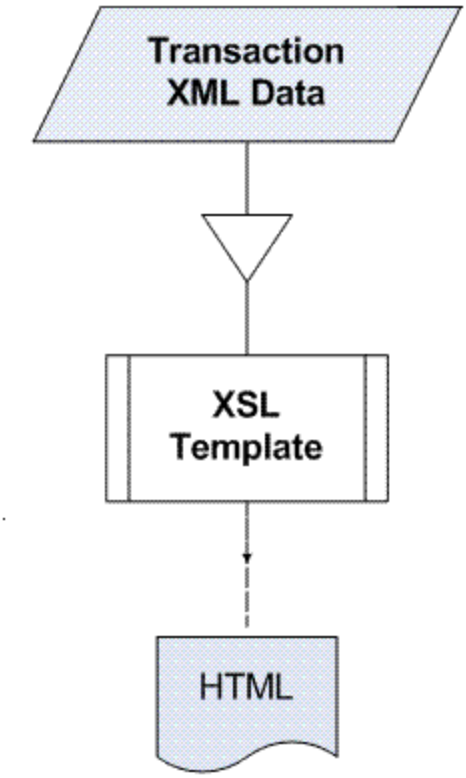
COMPOSITE ATTRIBUTE XSL VIEWS

OVERVIEW

Composite Attribute XSL views allow you to control the display of attributes from the transaction XML on the UI, depending on the XSL templates you upload.

You can create multiple XSL templates in this section to produce different types of HTML output on the transaction UI. Examples of composite attributes include configuration attributes and catalog attributes including model and product line attributes.

Composite Attribute XSL Overview



ADMINISTRATION

Adding a Composite Attribute XSL

1. Navigate from Admin Home Page > Commerce and Documents > Process Definition > Composite Attribute XSL Views from the Navigation drop-down list and click List.
2. Click **Add**.
3. Enter a **Name**, **Variable Name** and Description for the XSL View.
4. Click **Browse** and select an XSL file from your local computer.
5. Click **Add** to add the new file and go back to the page or Click Back to go return to the main page.

Previewing a Composite XSL View

1. Choose a document and a composite attribute type from the drop-down list, and click the Preview link or the Show XML link.
2. Click Display file to have a preview of the XSL file.
3. Enter a Transaction ID and click Load Transaction to preview how the XSL file will format actual data.
4. Do one of the following:
    - Click Apply to apply the changes made and refresh the page.
    - Click Update to update the changes made and return.
    - Click Back to go back to the main page.

RELATED TOPICS

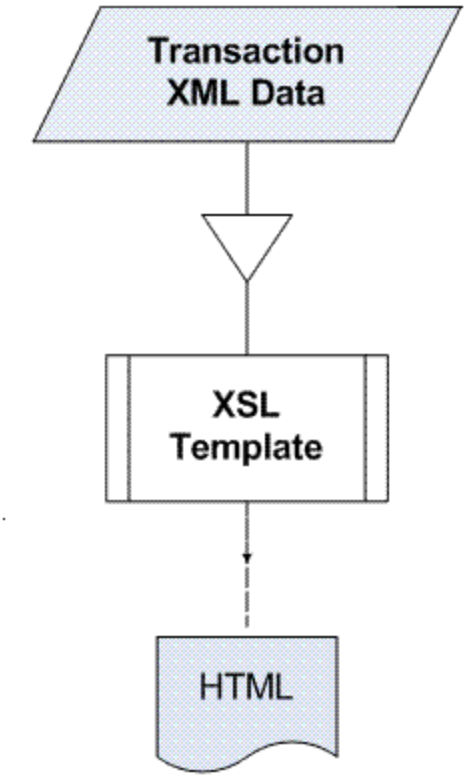
COMPOSITE ATTRIBUTE XSL VIEWS

OVERVIEW

Composite Attribute XSL views allow you to control the display of attributes from the transaction XML on the UI, depending on the XSL templates you upload.

You can create multiple XSL templates in this section to produce different types of HTML output on the transaction UI. Examples of composite attributes include configuration attributes and catalog attributes including model and product line attributes.

Composite Attribute XSL Overview



ADMINISTRATION

Adding a Composite Attribute XSL

1. Navigate from Admin Home Page > Commerce and Documents > Process Definition > Composite Attribute XSL Views from the Navigation drop-down list and click List.
2. Click **Add**.
3. Enter a **Name**, **Variable Name** and Description for the XSL View.
4. Click **Browse** and select an XSL file from your local computer.
5. Click **Add** to add the new file and go back to the page or Click Back to go return to the main page.

Previewing a Composite XSL View

1. Choose a document and a composite attribute type from the drop-down list, and click the Preview link or the Show XML link.
2. Click Display file to have a preview of the XSL file.
3. Enter a Transaction ID and click Load Transaction to preview how the XSL file will format actual data.
4. Do one of the following:
    - Click Apply to apply the changes made and refresh the page.
    - Click Update to update the changes made and return.
    - Click Back to go back to the main page.

RELATED TOPICS

PRINTER-FRIENDLY AND HISTORY XSL VIEWS

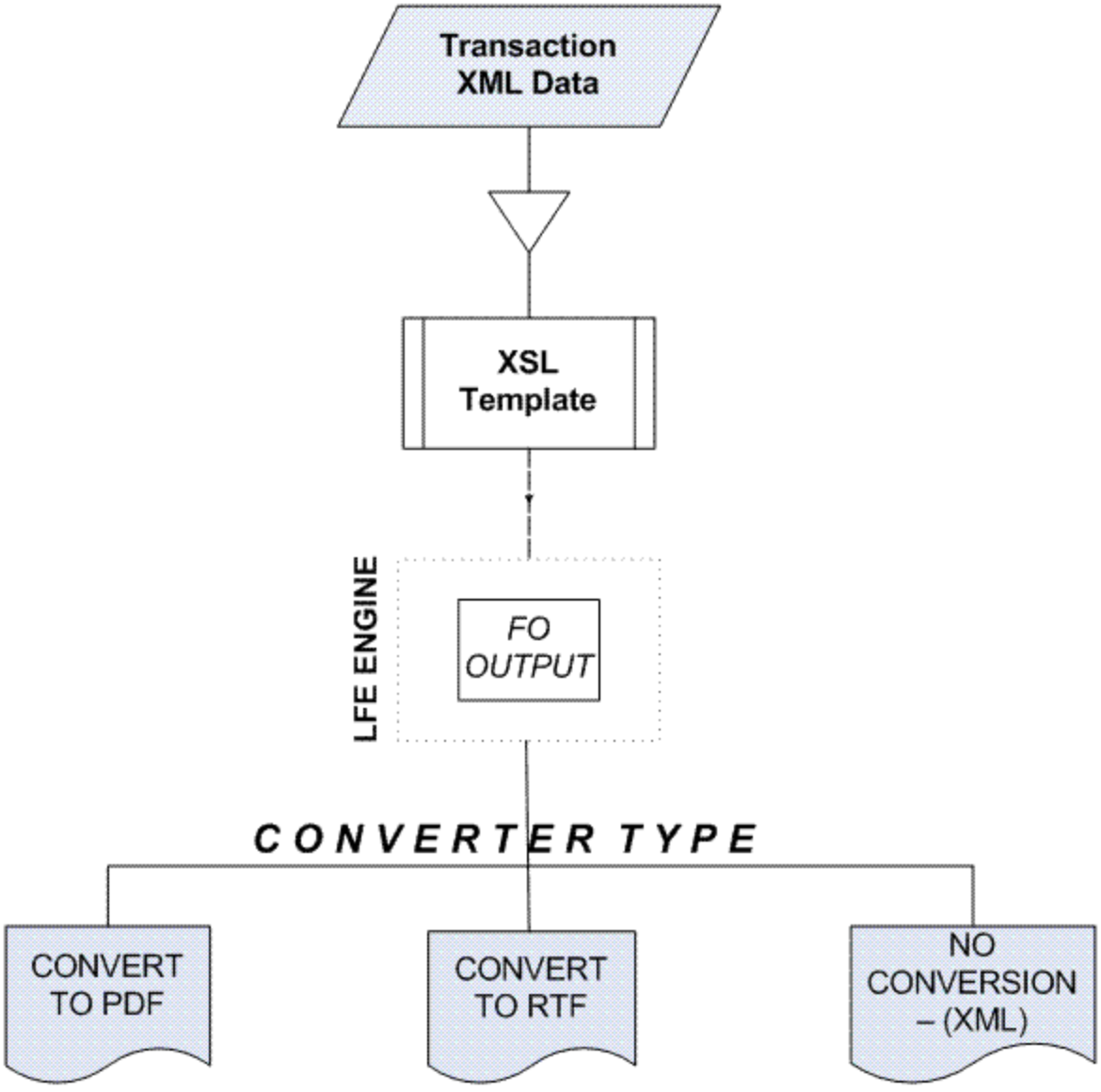
OVERVIEW

Printer Friendly/ History XSL views allow you to control the display of a transaction depending on the XSL template you select. You can create multiple XSL templates in this section to produce different types of printer friendly proposal formats or transaction history outputs.

These XSL templates can be predefined in the actions used to generate the printer friendly proposal or history output. Alternately, you can allow the user to select templates at run time to generate varied outputs.

You can generate ouput as XML, RTF, PDF, XLS, HTML, or DOCX.

Sample Workflow



ADMINISTRATION

Managing Printer-Friendly/History XSL Views

The Printer-Friendly/History XSL View page shows a list of XSLs that currently exist. Using this page, you can perform the following actions:

- **Delete XSLs:** Select all or a required XSL View Name that you want to delete using the Select All or Select options, and click Delete to delete it.
- **Re-order the XSL Views:**Modify the numeric value of the Order field. Click Apply to save and view the change made.
- **Update an XSL View:** Click an XSL View name to edit or update an XSL View. You can also enter a transaction ID and pre-view the XSL output from here.
- **Manage Translations:** Click the Translations button to enter translated text for the XSL View Names and Descriptions.
- **Add a Printer-Friendly/History XSL View**

Adding a Printer-Friendly/History XSL View

1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Printer-Friendly/History XSL Views from the Navigation Menu > List.
2. Click **Add**.
3. Enter a Name, Variable Name and Description for the XSL View Name.
4. Choose an **Output type** from the drop-down menu: None, XML, RTF, PDF, DOCX, XLS or HTML.

   If you have selected PDF then you can either lock the generated PDF or keep it unlocked. To lock the PDF output, check the Is PDF Read-Only box.

5. You can either use the BigMachines Document Engine to edit the XSLfile or upload a custom XSL.
   - **Editor:** Select this option if you are using the Document Engine, then choose the Selected Document. *Only deployed documents will be available for selection.*
   - **Upload XSL:** Select this option to upload a custom XSL File for the view. To do so, click Browse and select a file from your local system using the File Upload window that opens.



If you choose to upload a custom XSL, you must select the Commerce XML version. It is highly recommended that you choose the New XML version.

6. Click **Add** to save the view.



Adding a document and a printer-friendly XSL will NOT make your document available to the user. Remember to create a print action and link to the XSL View that you have created here.

Editing Printer-Friendly XSL Views

The XSL view modified on this page can be used to format print forms, email forms, and change history logs.

1. Edit any of the options in Steps 1-5 under Adding a Printer-Friendly XSL View.
2. Click Display file to have a preview of the XSL file.
3. Enter a Transaction ID to preview how the XSL file formats the actual data for Document, History, Document XML, and History XML.
4. Click Apply to apply the changes made and refresh the page. Click Update to update the changes made and go back to the page for managing printer friendly and history XSL views. Click Back to go back without saving.

DocX Documents

DocX documents are not available for use on iPad. It is possible to view RTF files if the file has been emailed to someone via an e-mail action. The steps below use the build-in mail application, Mail, which is standard on the iPad.



Disclaimer: This is what we tested with. We are not responsible for the functionality of any 3rd party application.

1. Go to the App Store and download the free app doctape
2. Go to Mail and open the e-mail containing the document you wish to view
3. Press and hold the attachment icon until a box pops up containing the options ‘Quick Look’ and ‘Open in “doctape”’.

4. Select ‘Open in “doctape”’. The app will upload the attachment then take you to a list of all documents opened through doctape. The one you just uploaded will be at the top.

5. Select the document you want to open by pressing its icon

6. In the Mail app, if you just click the attachment or select ‘Quick Look’ through the popup options, a new screen will open but it will just be a blank white page

    - If this happens, you can simply click the action button in the upper-right corner (it looks like a box with a curved arrow pointing to the right) and then select ‘Open in “doctape”’

7. Free apps that were tested and found to be incompatible:

    - OliveOfficeBasicHD
    - FileApp
    - PlainText
    - SimpleNote

RELATED TOPICS

WORKFLOW STEPS

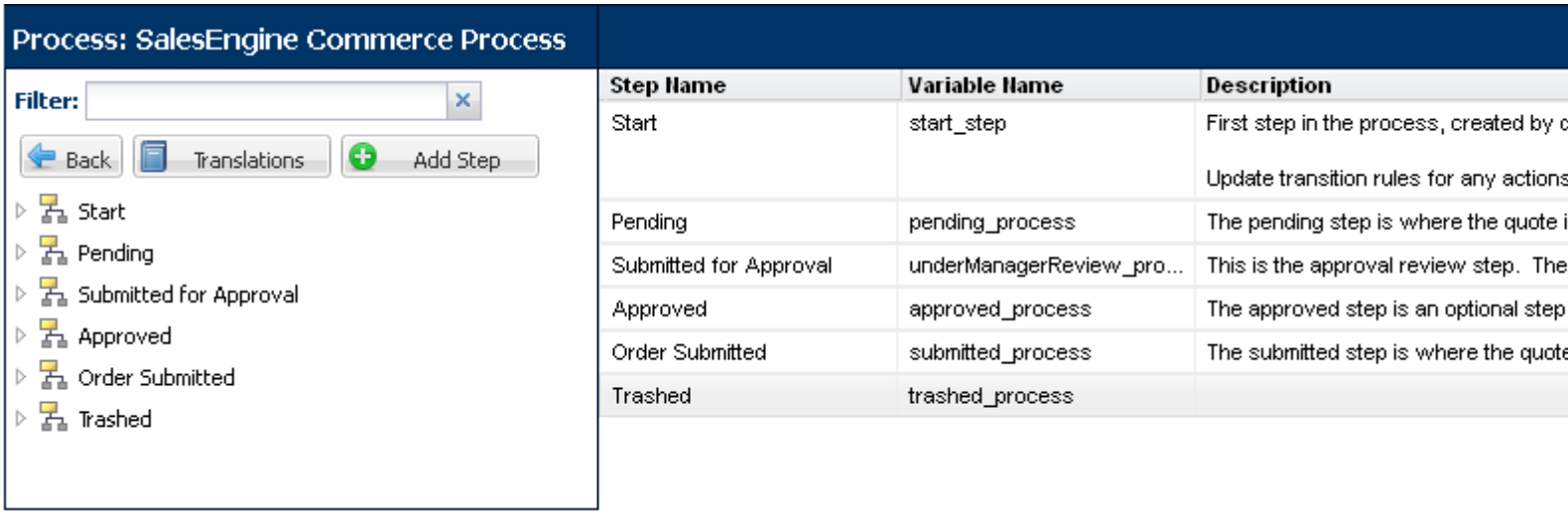
OVERVIEW

Workflow administration is the final step in setting up a commerce process. Workflows are made up of steps, which define the states transactions can be in, as well as document permissions and routing. Commerce processes can have any number of workflow steps. For example, an RFQ process could have steps like "Submitted", "Quoted", "Accepted", "Declined" and "Expired". These steps could transition a transaction from an RFQ document to a Quote document to a PO document.

Workflow steps use profiles to define access rights, transition notification, and transaction views. The commerce system automatically creates a default profile for each workflow step. You can customize the default profile and create additional ones to support different transaction access rights. Profile permissions can be granted based on user access type, user groups or previous performers. In addition to these permissions, you can add auto-forwarding rules to workflow steps to support a collaborative sales environment, one in which multiple users can work on the same transaction. Auto-forwarding rules direct the system to share transactions between members of certain user groups. You can create auto-forwarding rules for each workflow step and base them on any number of criteria.

All workflow step functionality can be administrated in commerce processes that have not been deployed. After a process has been deployed, certain functionality becomes hidden from view, including adding, ordering and deleting steps. To perform any of these actions when they are not visible, contact a BigMachines implementation engineer.

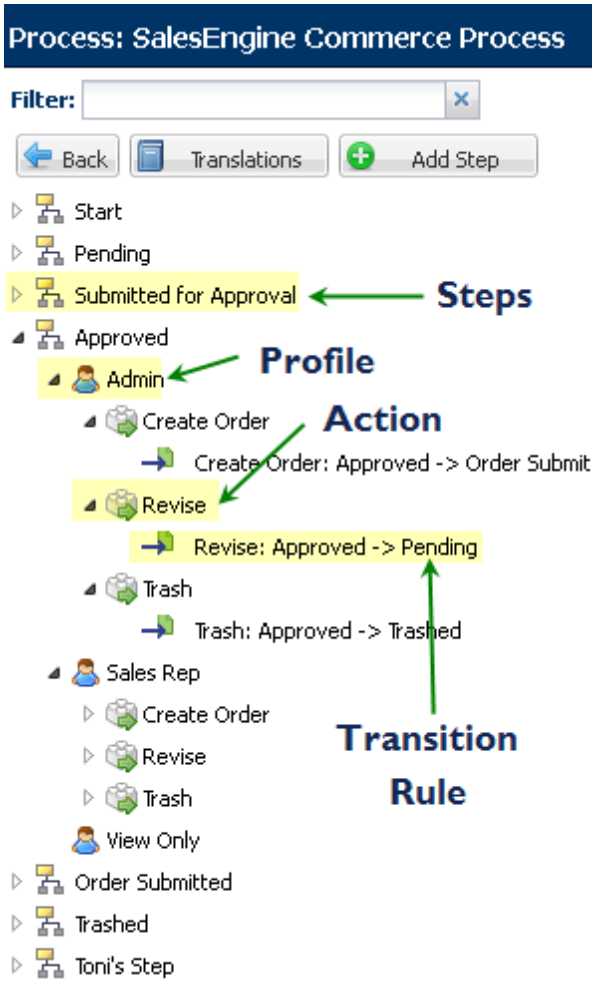
Example of One-Page Step Administration



ADMINISTRATION

You can add/edit and delete all steps, profiles, transition rules and timers from a single page. Steps and profiles are organized in a hierarchical tree structure for ease of administration. You can re-order your steps and profiles and move transition rules using drag-and-drop.

How it's all related

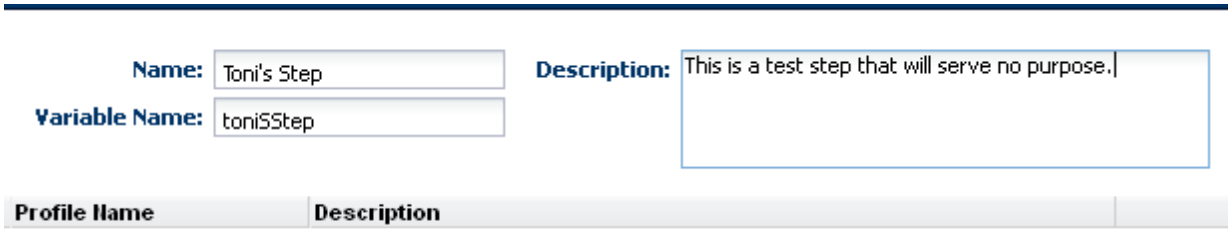


Creating Workflow Steps

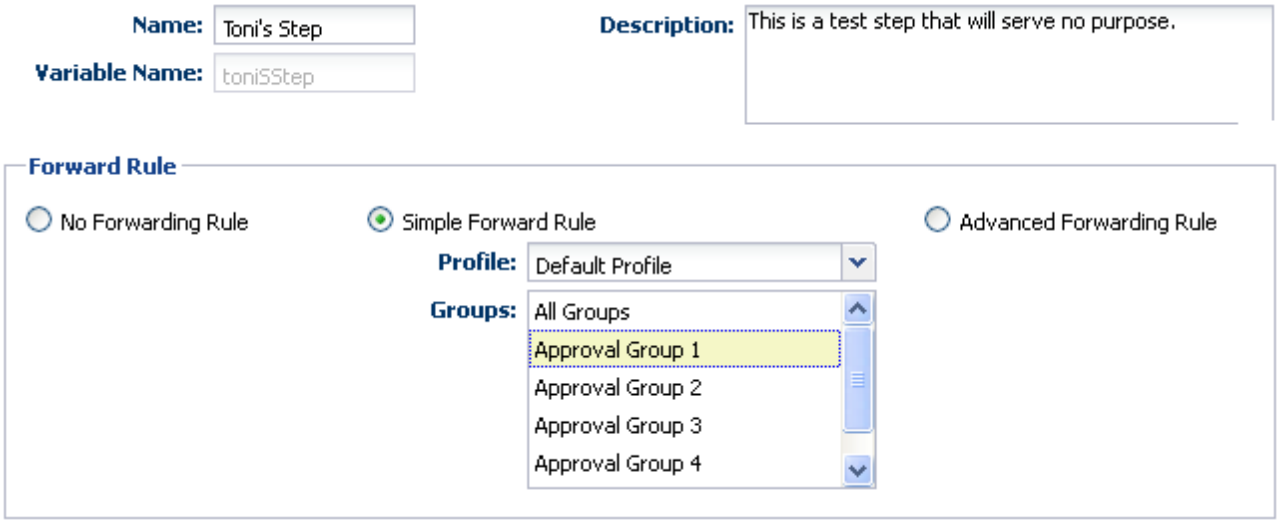
Steps (document states) can be created individually from the Step List page. We recommend creating all of the steps in a workflow before managing profiles.

1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps.
2. Enter a step **Name and Variable Name**. *The name is displayed when you encounter the step. For example, Start.*
3. Enter a **Description** for the step. For example, Default Step.
4. Click **Save**.

This displays the section where you can define forwarding rules for the step. Forwarding rules are used to give one or more user groups access to transactions during that step. Note: To understand the concept of auto-forwarding rules, please see Auto-Forwarding - An Overview.



5. Specify the following in the **Forward Rule** section:
   - Select **No Forward Rule** if you do not want to add a forwarding rule. This is the default section.
   - Select **Simple Forwarding Rule** if you want to add a forwarding rule that doesn't require BML.
   - Select a **Profile** from the drop-down list provided.
   - Choose **All Groups** if you want to link this step to all the groups or select individual group names if you want to link this step to a few selected groups. Select multiple group names using the CTRL key from the names provided in the list box.
   - Select **Advanced Forward Rule** if you want to add an advanced forwarding rule and click Define Function. This opens the Commerce BML Editor window.



6. Click **Save**. *The new step appears in the right panel.*



To edit workflow steps, double-click on the Step Name in the right pane.



Steps are created in an inactive, pending state. After all components of a commerce process have been completed, the process should be deployed. For more information, see Deploying Processes.

NOTES



Use the hidden **input_step_varname** in the transaction HTML to find out the step of the transaction.

IDs and descriptions for all steps can be viewed easily after accessing the workflow page. You can view them by clicking on a Step name.

You can use the Filter feature to dynamically drill down through the Steps, Profiles, Actions and Rules. For example, you could type "Sales" into the Filter, and any Step Profile, Action or Transition Rule that has "Sales" in the name would dynamically display, improving the speed of maintenance.

When performing a Mass Update, users can filter which steps the update applies to.

RELATED TOPICS

WORKFLOW STEPS

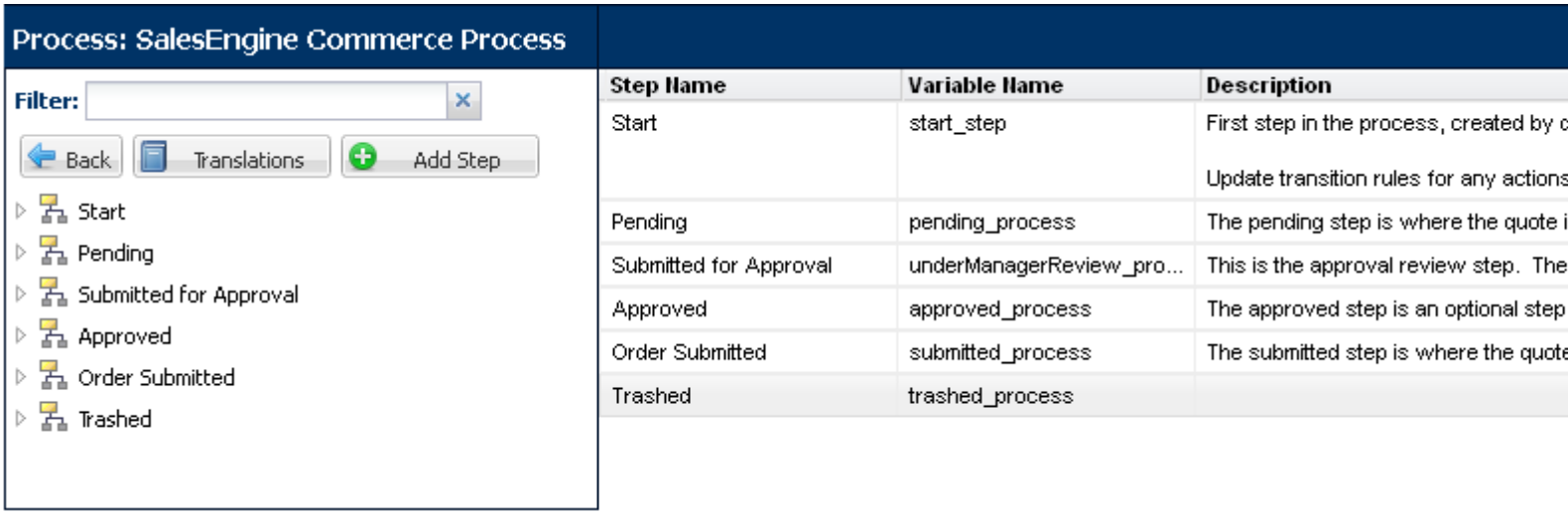
OVERVIEW

Workflow administration is the final step in setting up a commerce process. Workflows are made up of steps, which define the states transactions can be in, as well as document permissions and routing. Commerce processes can have any number of workflow steps. For example, an RFQ process could have steps like "Submitted", "Quoted", "Accepted", "Declined" and "Expired". These steps could transition a transaction from an RFQ document to a Quote document to a PO document.

Workflow steps use profiles to define access rights, transition notification, and transaction views. The commerce system automatically creates a default profile for each workflow step. You can customize the default profile and create additional ones to support different transaction access rights. Profile permissions can be granted based on user access type, user groups or previous performers. In addition to these permissions, you can add auto-forwarding rules to workflow steps to support a collaborative sales environment, one in which multiple users can work on the same transaction. Auto-forwarding rules direct the system to share transactions between members of certain user groups. You can create auto-forwarding rules for each workflow step and base them on any number of criteria.

All workflow step functionality can be administrated in commerce processes that have not been deployed. After a process has been deployed, certain functionality becomes hidden from view, including adding, ordering and deleting steps. To perform any of these actions when they are not visible, contact a BigMachines implementation engineer.

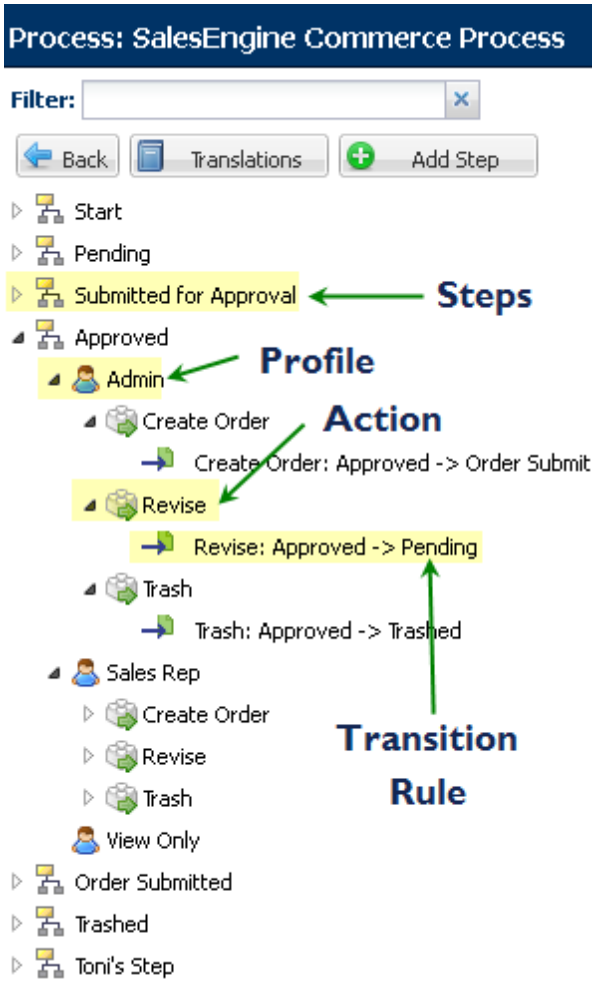
Example of One-Page Step Administration



ADMINISTRATION

You can add/edit and delete all steps, profiles, transition rules and timers from a single page. Steps and profiles are organized in a hierarchical tree structure for ease of administration. You can re-order your steps and profiles and move transition rules using drag-and-drop.

How it's all related

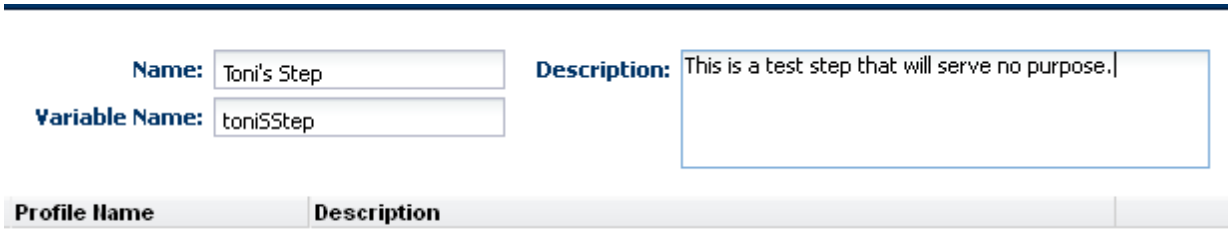


Creating Workflow Steps

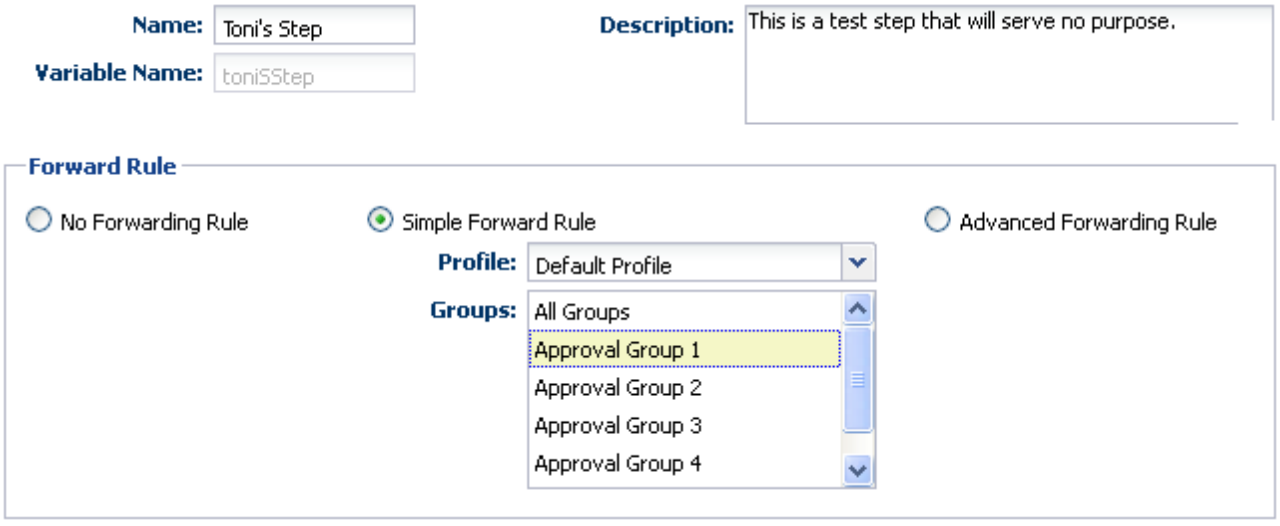
Steps (document states) can be created individually from the Step List page. We recommend creating all of the steps in a workflow before managing profiles.

1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps.
2. Enter a step **Name and Variable Name**. *The name is displayed when you encounter the step. For example, Start.*
3. Enter a **Description** for the step. For example, Default Step.
4. Click **Save**.

This displays the section where you can define forwarding rules for the step. Forwarding rules are used to give one or more user groups access to transactions during that step. Note: To understand the concept of auto-forwarding rules, please see Auto-Forwarding - An Overview.



5. Specify the following in the **Forward Rule** section:
   - Select **No Forward Rule** if you do not want to add a forwarding rule. This is the default section.
   - Select **Simple Forwarding Rule** if you want to add a forwarding rule that doesn't require BML.
   - Select a **Profile** from the drop-down list provided.
   - Choose **All Groups** if you want to link this step to all the groups or select individual group names if you want to link this step to a few selected groups. Select multiple group names using the CTRL key from the names provided in the list box.
   - Select **Advanced Forward Rule** if you want to add an advanced forwarding rule and click Define Function. This opens the Commerce BML Editor window.



6. Click **Save**. *The new step appears in the right panel.*



To edit workflow steps, double-click on the Step Name in the right pane.



Steps are created in an inactive, pending state. After all components of a commerce process have been completed, the process should be deployed. For more information, see Deploying Processes.

NOTES



Use the hidden **input_step_varname** in the transaction HTML to find out the step of the transaction.

IDs and descriptions for all steps can be viewed easily after accessing the workflow page. You can view them by clicking on a Step name.

You can use the Filter feature to dynamically drill down through the Steps, Profiles, Actions and Rules. For example, you could type "Sales" into the Filter, and any Step Profile, Action or Transition Rule that has "Sales" in the name would dynamically display, improving the speed of maintenance.

When performing a Mass Update, users can filter which steps the update applies to.

RELATED TOPICS

DOCUMENT VIEWS

OVERVIEW

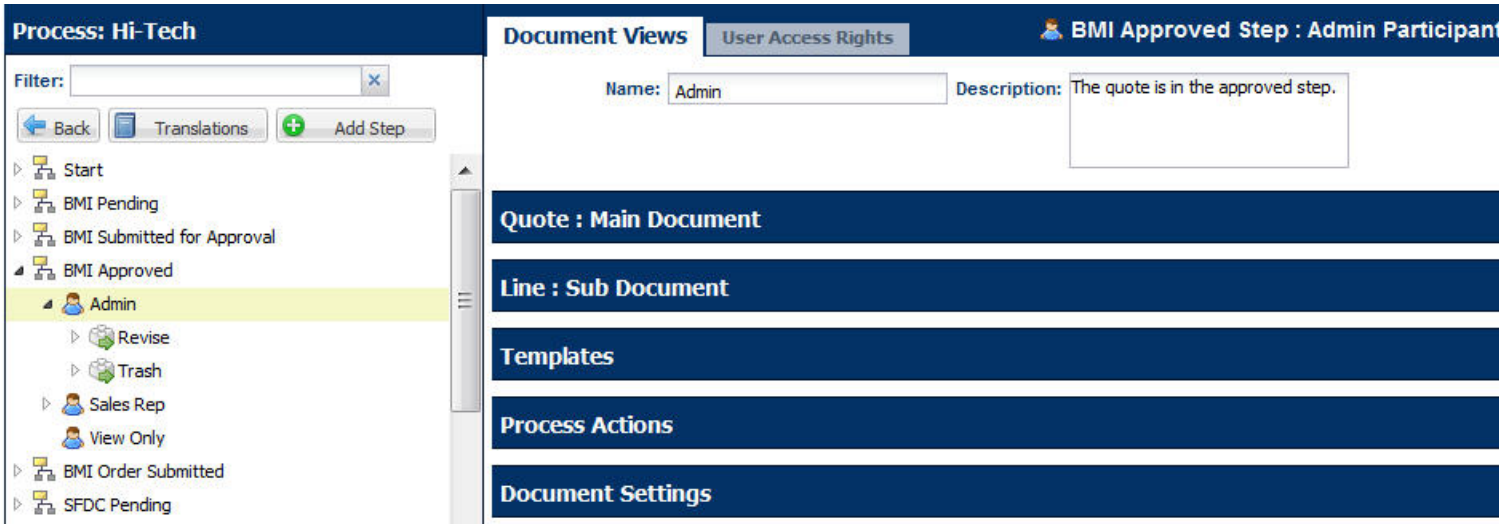
Document views set permissions for which document attributes, document actions and process actions are active in a profile. They control whether document attributes are read-write, read-only or hidden. Views also control whether document actions are active or inactive. Active actions appear to users, inactive ones are hidden from view.

For example, you might want to give a supplier read-write access to "ship date" field, whereas for buyers, you might just want to display the field, but not let them edit it. To manage two different types of behavior for the same attributes in a document, you would use two different profiles, each with its own profile view.

ADMINISTRATION

Managing Document Views

1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps
2. Click on a **desired step** to expand it.
3. Double-click the **participant profile** whose Document Views need to be set or revised.
4. Click on an **individual section** to expand and set permissions:
    - **Quote: Main Document:**
        - **Tabs:** Select Show or Hide for each main-document tab.
        - **Actions**: Select Active, Inactive or Hide for each main-document action.
        - **Attributes:** Select Read/Write, Read-Only or Hide for each main-document attribute.
    - **Line: Sub Document:**
        - **Tabs:** Select Show or Hide for each sub-document tab.
        - **Actions**: Select Active, Inactive or Hide for each sub-document action.
        - **Attributes:** Select Read/Write, Read-Only or Hide for each sub-document attribute.
    - **Templates:**Select Show or Hide for each Rich Text Editor Template.
    - **Process Actions:** Select Active, Inactive or Hide for each of the commerce process actions.
    - **Document Settings:** Select Show or Hide.



5. Click **Save**.

RELATED TOPICS

TRANSITION RULES

OVERVIEW

What are they?

Transition rules designate when the next document or document state triggers. Actions that do not take users to a different document (or document state) do not require transition rules.

How do they work?

Within transition rules you can also set up email, fax, or XML notification to trigger when an action occurs. Transitions from one step to another will also trigger auto-forwarding rules that have been set up for the step a transaction moves into.

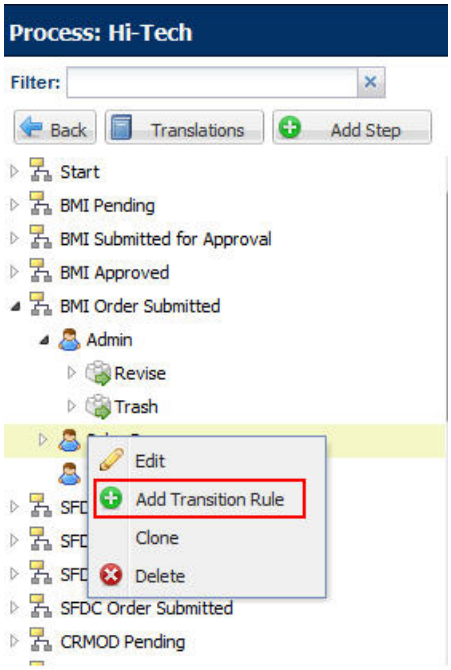
Here are some rules regarding transition rules:

- They are bound to one and only one action.
- They can change the step in the commerce process.
- If a user does not belong to a profile in the next step, the user loses all access rights to the process.
- You can order transition rules to set rule precedence.
- When the commerce system can trigger more than one rule for an action in a single step, the system triggers the transition rule with the highest precedence.

ADMINISTRATION

Creating a Transition Rule (Moving Docs between Workflow Steps)

1. Navigate from: Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps
2. Click on a **desired step** to expand it.
3. Right-click on a **participant profile.**
4. Click **Add Transition** Rule.

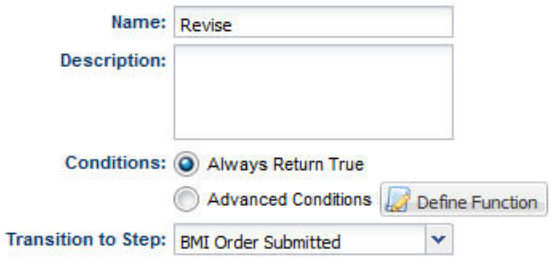


5. Enter a **Name and Description** for the rule.
6. Choose a **Process or Quote action** from the Action drop-down menu. *This is the action that will trigger the transition rule.*
7. Click **Save**.



This will remove the Action drop-down menu and replace it with condition and transition options.

8. Select the rule **Conditions**:
   - **Always Return True**: The transition rule will always fire.
   - **Advanced Conditions**: Click Define Function to create the conditions in which the rule will fire through a BML script.
9. Choose the **step that the document will be moved to** from the Transition to Step drop-down menu.



10. Click **Save**.
11. Click **Define Notifications** to manage any notifications that are sent when a transition rule triggers.

NOTES



Transition rules are tied to profiles and actions, rather than timers exclusively. This means, a transition rule will trigger if any user in the profile performs the document action.

RELATED TOPICS

FORWARDING RULES

OVERVIEW

What Are They?

Auto-Forwarding enables you to set up collaborative transaction handling. Auto-forwarding rules should be set up for every workflow step, if you want to create a collaborative sales environment from the beginning of a transaction to the end. The benefit of setting up auto-forwarding is that it happens in the background and guarantees that documents become available to groups of users as soon as the document transition occurs.

How Do They Work?

Whenever a document moves from one step to another, the system can fire auto-forwarding rules and distribute transactions to one or more user groups or, to a specific user. Documents can be forwarded to specific groups, or to all groups that the current user belongs to. Any user or user in a group that receives a forwarded document can then work on the transaction. Individual access rights do not restrict users from accessing auto-forwarded transactions.

Before auto-forwarding rules are created, make sure you've created the necessary user groups.

Permissions to document attributes and actions for auto-forwarded documents are based profile view of the participant profile associated with the auto-forwarding rule. Updates made to auto-forwarded documents become visible to the user groups who received the auto-forwarded transaction.

Types of Forwarding Rules

Simple

Simple auto-forwarding rules are defined with an associated participant profile, one or more user groups or, a specified user.

In auto-forwarding rules, associated profiles set which view is used for forwarded transactions. Profile views designate whether actions and attributes are hidden or visible for the user groups that receive the forwarded document. In some cases, a user might have access to more than one profile. This can happen if a user is included in an auto-forwarding group and also has been granted specific user access rights to a different profile. When this happens, the system displays the view of the highest ranked profile. Profile ranking is based on the order of profiles in a given step.

In simple rules, associated user groups can be set to "All Groups", meaning all groups that the action performer belongs to, or "Selected Groups", meaning a hard-coded set of supplier groups which the supplier user may or may not belong to. For the "Selected Groups" option, transactions are only forwarded to the groups the supplier user belongs to.



When a simple auto-forwarding rule is set up for a step using the "All Groups" option, the rule will work for buyer users if a buyer user who performs an action belongs to a buyer company group. Buyer groups are not available for selection in the Selected Groups list box though, so auto-forwarding will not work for buyers when a rule is defined using "Selected Groups".



The system does not restrict forwarding based on the user's access type. Whoever is in a user group set up to receive forwarded documents is granted access to forwarded transactions. The document view for all users in the group is based on the forwarding rule's associated profile.

Advanced Rules

Advanced auto-forwarding rules enable you to set up conditions for forwarding. Using advanced rules, you can instruct the system to forward documents to one group or another based on some condition of an order, such as the price, date, or any other document attribute.

There is a variable created specifically for auto-forwarding called "_system_user_groups". This variable contains the current user's groups, which can be assigned to a single participant profile in auto-forwarding scripts. The "_system_user_groups" variable works for both buyer user groups and supplier user groups.

Format of Return Value: group~company~participantProfile [|group~company~participantProfile]*

Example Advanced Script

```
/**
* FORWARDING RULE
* company - represents the buy access company name
* group - set to a static value 'sales_rep'
* pp - represents participant profile also set to a static value
*/
company = quote_sender;
group = "sales_rep";
pp = "Resellers_Sender_Submitter";

result = "";
if ( companyLoginName <> _system_supplier_company_name )
{
result = group + "~" + company + "~" + pp;
}

return result;
```

ADMINISTRATION

Please see Workflow Steps for information on how to create a forwarding rule.

AUTO-FORWARDING USE CASE

Auto-Forwarding functionality uses groups to define who should have access to transactions. To demonstrate how simple auto-forwarding rules work, here are two sample groups and the result of three possible scenarios.

Western Agents Group

Members: SuperUser, Andy, Fred, Pete

Eastern Agents Group

Members: superuser, Andy, Gary, Tom

The following table shows which groups would have access to transactions whenever a document transitions into a step that has simple auto-forwarding rules set up to distribute documents to "All Groups", meaning all groups to which the current user belongs.

Document Performer	Western Group	Eastern Group
SuperUser or Andy saves an order	x	x
Fred or Pete Saves an order	x	
Gary or Tom saves an order		x

All users in the groups that receive the forwarded documents have equal access to the transaction, regardless of user type. Access to forwarded transactions is based only on the profile associated with the forwarding rule.

NOTES



Auto-Forwarding rules should be created after steps and profiles have been created.

Permission to auto-forwarded transactions is granted to all members of the user groups associated with an auto-forwarding rule.

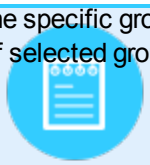
Delimiters (~, + and @) that are part of advanced auto-forwarding rules, cannot be used in group names. Group names can only contain alphanumeric characters and they must use underscores for spaces. Also, names cannot begin with an underscore.



The participant profile selected for the group defines the document view users in the group will see. If a user is associated with multiple participant profiles, via either an auto-forwarding rule or profile users access rights, then the view displayed is based on the highest ranked participant profile. Participant profiles can be ordered to set their ranking, see Ordering profiles.

"All Groups" means all groups that the current user is a member of. This type of forwarding works for both supplier company groups and buyer company groups. "Selected Groups" enables you to hard code

the specific groups you want transactions to be forwarded to. The current user has to belong to the set of selected groups. It's possible to choose selected groups for supplier company groups only.



RELATED TOPICS

PARTICIPANT PROFILES

OVERVIEW

What are they?

Participant profiles are used to create different rules for different user types. Commerce will always generates a default profile whenever a new workflow step is generated. In addition to the default profile, you can also create custom profiles.

How do they work?

There are two tasks involved in profile administration:

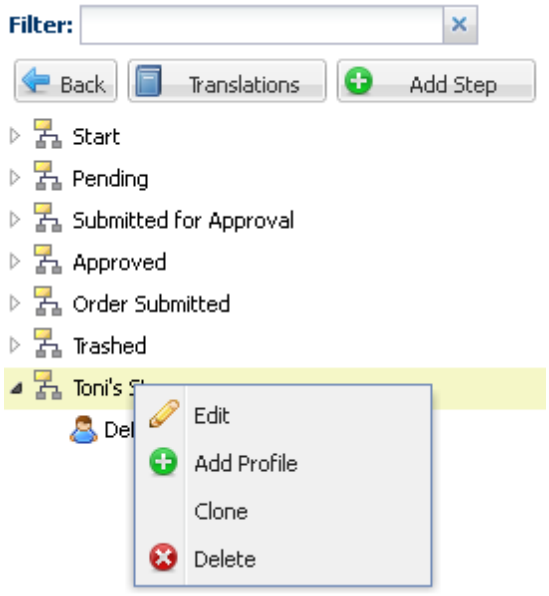
1. Create the profile name. The system performs this automatically for the default profile.
2. Define the document views, user access rights and transition rules.

When a workflow step contains multiple profiles, you can use drag & drop functionality to order them to set the profile precedence. BigMachines uses profile precedence to determine which profile to use when a user is included in more than one within a single step.

ADMINISTRATION

Managing Participant Profiles

1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps
2. Right-click on **name of the Step** you'd like to add the profile to.
3. Click **Add Profile**.



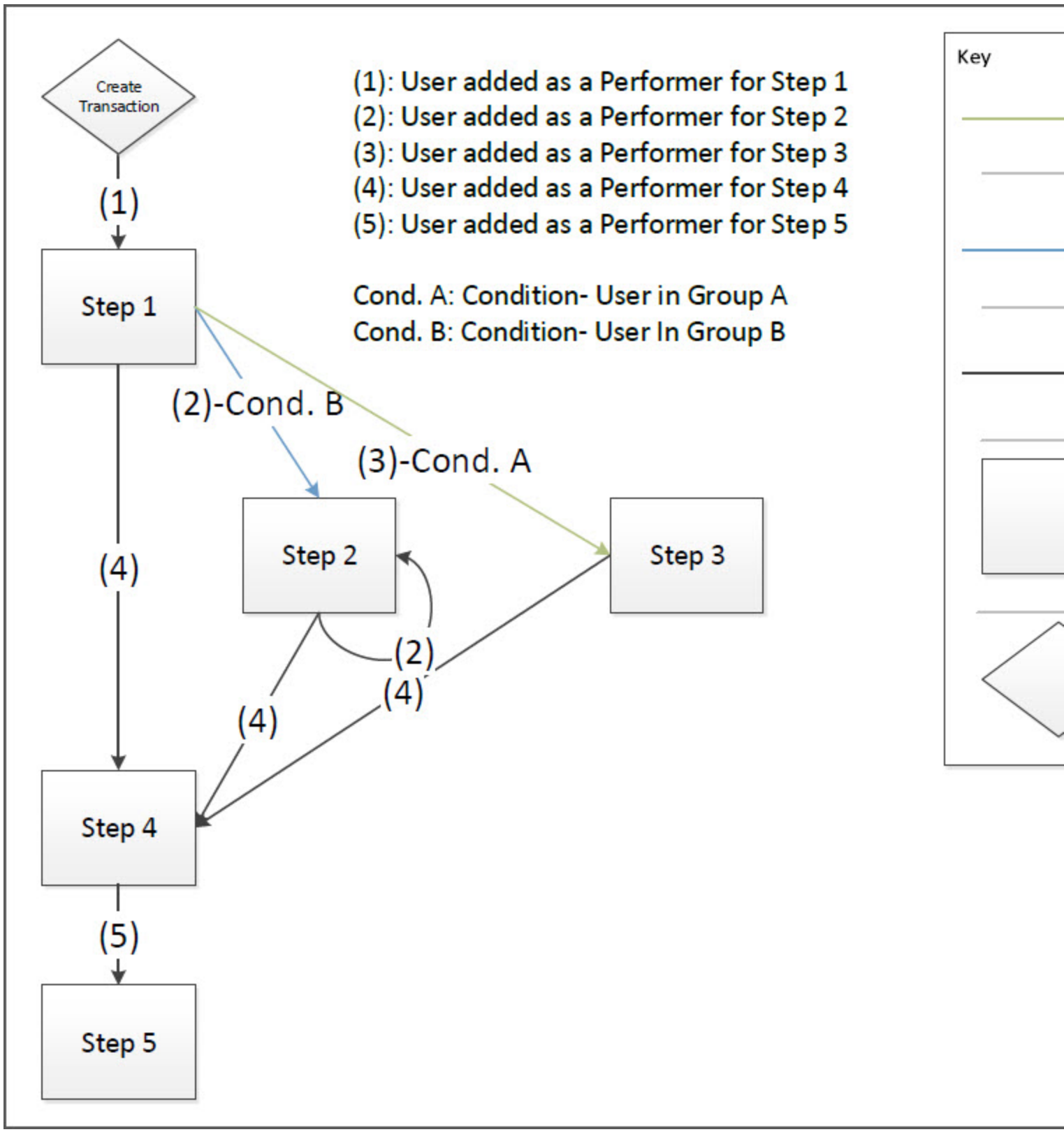
4. Enter a **Name and Description**for the profile on the Document Views tab.

5. Click the User Access Rights tab to select which users have access to the profile.
6. Click **Save**.

Managing Performer Steps

Performers of a given step are users that fired a transition into that step. A user cannot get access to a destination step based on being a performer of that step unless previously established (not the same transition).

Process Setup


Create Transaction
(1)
Step 1
(2)-Cond. B
(3)-Cond. A
Step 2
Step 3
(4)
(2)
(4)
(4)
Step 4
(5)
Step 5
(1): User added as a Performer for Step 1
(2): User added as a Performer for Step 2
(3): User added as a Performer for Step 3
(4): User added as a Performer for Step 4
(5): User added as a Performer for Step 5
Cond. A: Condition- User in Group A
Cond. B: Condition- User In Group B
Key

Step	Participant Profile: *Function*
Step 1	**Admin:** *Group A* **Sales Rep:** *All Company Types/Groups*
Step 2	**Internal Sales:** *Group A* **External Sales Rep:** *Performer for Step 1*
Step 3	**Internal Sales:** Performer for Step 1
Step 4	**Internal Sales:** *Performer for Step 3* **External Sales Rep:** *Performer for Step 2* **View Only:** *Performer for Step 1*
Step 5	**Internal Sales:** *Performer for Step 3* **External Sales Rep:** *Performer for Step 2* **View Only:** *Performer for Step 1*



In order to gain access to a Step via the Performer Step, the user gaining the access must be considered a Performer for the Step defined in the Performer Step.

NOTES



Select a profile you want to create a duplicate of, right-click and click Clone. A Copy of <selected profile> gets added to the list of profiles. You can clone only one profile at a time. Cloning a profile makes a copy of all sub-components including transition rules.



Profiles are created in an inactive, pending state. After all components of a commerce process have been completed, the process should be deployed.



It is not possible to recover the deleted data. Upon a deletion, all associated profiles and profile rules are also deleted.



Delimiters (~, + and @) that are part of advanced auto-forwarding rules, cannot be used in profile names. Profile names can only contain alphanumeric characters and they must use underscores for spaces. Also, names cannot begin with an underscore.

RELATED TOPICS

TIMERS

OVERVIEW

What are they?

You can schedule the system to automatically perform a document action by using Timers. Based on a defined duration of time, a Timer triggers when the elapsed time exceeds the specified duration. BigMachines uses date type attributes to calculate elapsed time and Timers are set up on modify type commerce actions.

How do they work?

Timers are managed within individual workflow steps and you can define one each. Each Timer is associated with a participant profile and a user account.

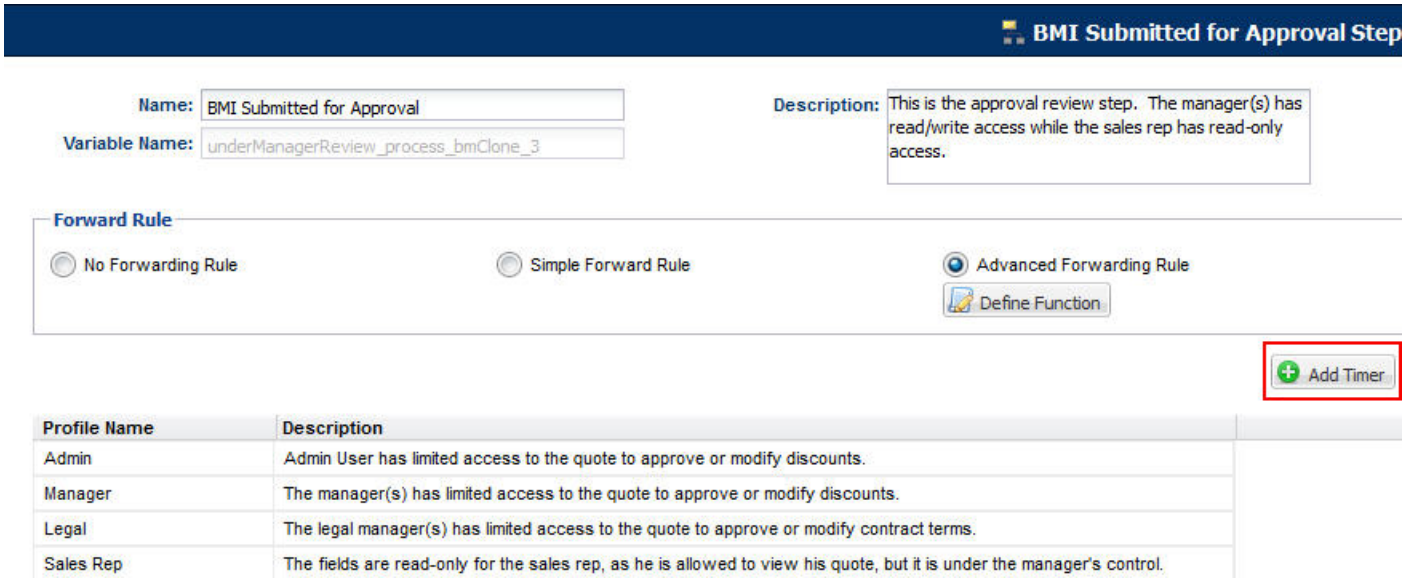
- The user account acts as the performer of the timed action and is automatically granted permission to view the document.
- The participant profile defines whether the timer has permission to perform the action. In other words, the action you want a timer to trigger must be active in the profile associated with the timer.

Using transition rules in a timer's associated profile, you can configure the system to send email notifications when a timed system triggers. You can also use transition rules to transition a workflow step from the current step to any downstream step when a timed action is performed.

ADMINISTRATION

Adding a Timer

1. Navigate from: Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps
2. Double-click on a **step name** in the left panel. If you need to create or edit workflow steps, click here.
3. Click **Add Timer**.



4. Enter a **Name and Variable name** for the timer.
5. Choose a **Participant Profile** to map the timer to from the Map to Profile drop-down menu.

> This is the profile with which the application executes the timer action. Since timers are used in transition steps, and transition rules are tied to the junction of actions and profiles, by mapping the timer action to a particular profile, the admin can specify which transition rules will operate.

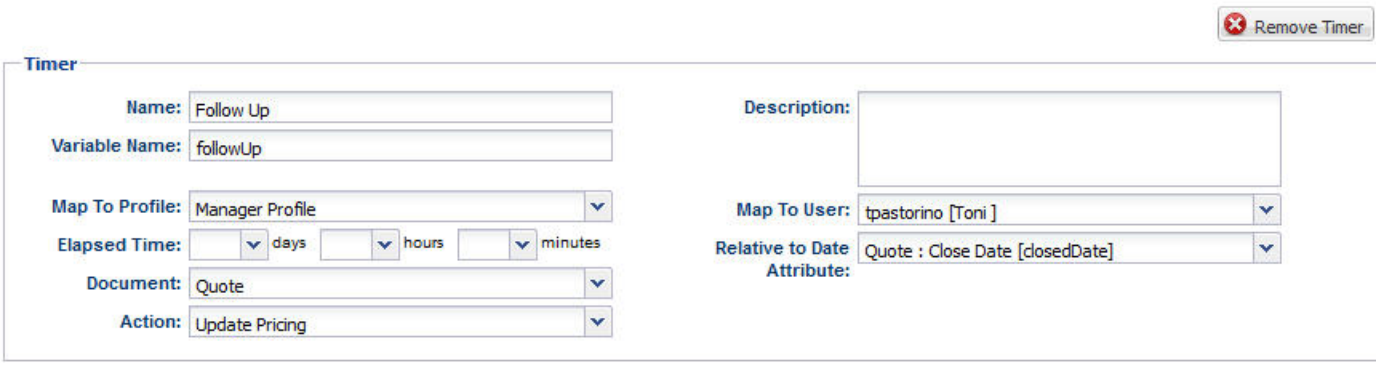
6. Choose a **user** to map the timer to from the Map to User drop-down menu.

> This is the user that the application 'proxies' to accomplish the named action when the timer executes. This mapping allows the fullaccessuser to specify 'who' initiated the action.

7. Define the **duration of time** for the timer by specifying the time to elapse in the Elapsed Time field. Choose the Days, Hours and Minutes in the drop-down lists provided.
8. Choose a **date attribute** you want to relate to the timer from the Relative to Date Attribute drop-down list provided.

> The Relative to Date Attribute drop-down list contains commerce Date Type attributes that have been added to the data columns. To view a commerce date attribute in this list, add the date attribute to the data columns.

9. Choose a **document** from the Document drop-down list provided.
10. Choose an **action** from the Action drop-down list provided.



11. Click **Save**. *This displays the section where you can define action rules for the step.*
12. Select an **action rule** as Simple or Advanced.
    - Select Advanced if you want to add an advanced action rule and click Define Function.
13. Click **Save**.

USE CASES

Use Case: Recurring Timers

1. Create a **date attribute**. *It can be set to system time, system timer plus some interval (e.g. 30 days from now), or a given time (e.g. end of calendar year).*



The date attribute must be included in the data columns or else it cannot be used to trigger a timer.

2. The timer should trigger an action that modifies your new date attribute. It could set to the current time, or another interval, or a new given time. **Example:** if you want to use a timer to send a reminder about open quotes every two weeks, the action should set the date attribute to system time plus two weeks.

TROUBLESHOOTING

Recurring Timer still fires after quote exits "timer-ed" step

1. Make the "lastEmailed_quote" Attribute (or whatever is controlling the Recurring Timer) a writable Attribute for administrators.



If the problem is associated with only one quote, set the date to sometime in the far future.

2. If more than one quote is the issue, double-check your Timers. *The above fix does not work for a systematic problem.*

NOTES



The system records who performs document actions, including timer actions, in the change history log. You can create a virtual user solely for the purpose of performing timer actions, see Setting up a timer user.

To add email notification or a step transition rule functionality to a timer, you can add transition rules to the profile associated with the timer. For more information, see Managing Notifications.



The user account specified for a timer is recorded by the system as the performer of the timer action. This performer appears in change history logs for documents. You can select an existing user account as the performer of a timer action, or you can create a user account specifically for the purpose of performing timer actions.

Timers can perform actions up to 99 days after the date defined by the date field.

Timers are created in an inactive, pending state. To activate a timer for all documents, including legacy documents, deploy the commerce process.



A Timer Action is always performed AFTER THE DATE specified by the Relative to Date Attribute. A timer cannot be set up to occur before a date.

RELATED TOPICS

MOBILE APPROVALS

OVERVIEW

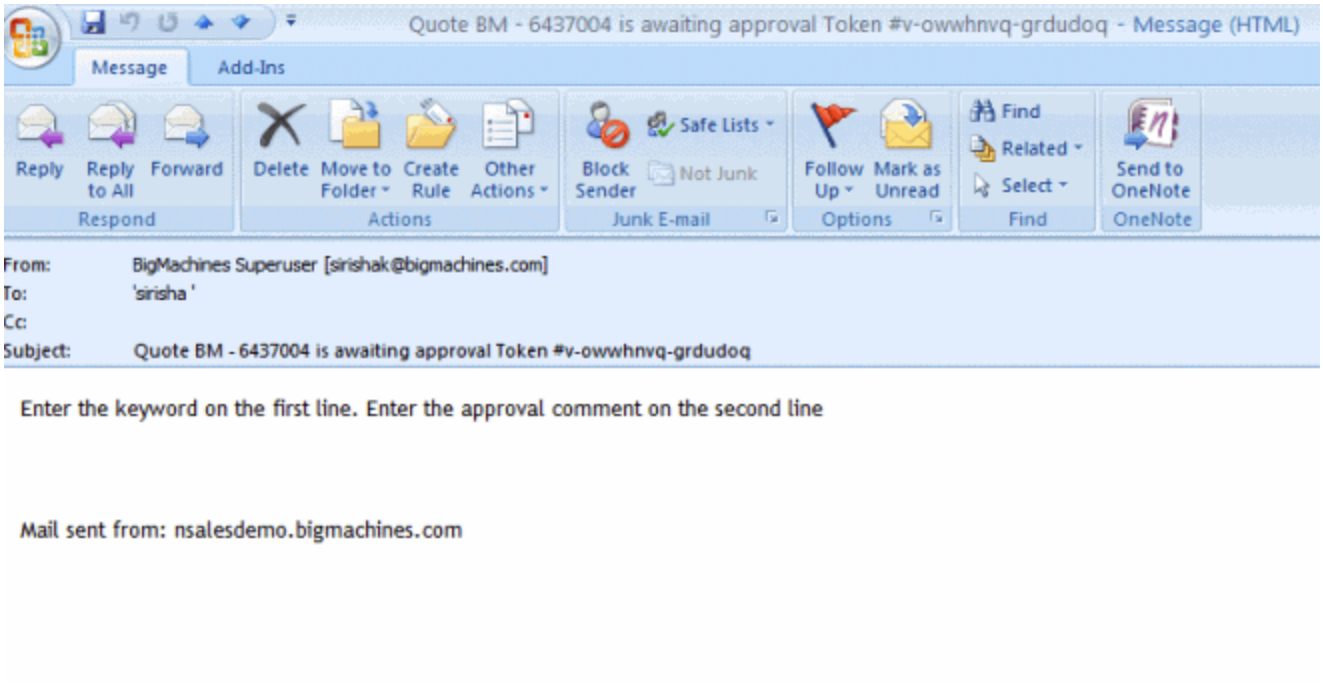
What are they?

This feature allows users to perform modify actions on a quote by replying to an email. The typical use case for this is to allow a user to approve/reject a quote by responding to a system generated email from a mobile device. The user can also provide a comment in the email that can be mapped to a history or text area type attribute. For more information on this functionality, watch this demo.

How do they work?

When a BigMachines registered user performs a step change action then a notification is sent to the user. If the Mobile Devices approvals process is set up on this notification rule, then the email sent to the user will contain a unique key in the subject line.

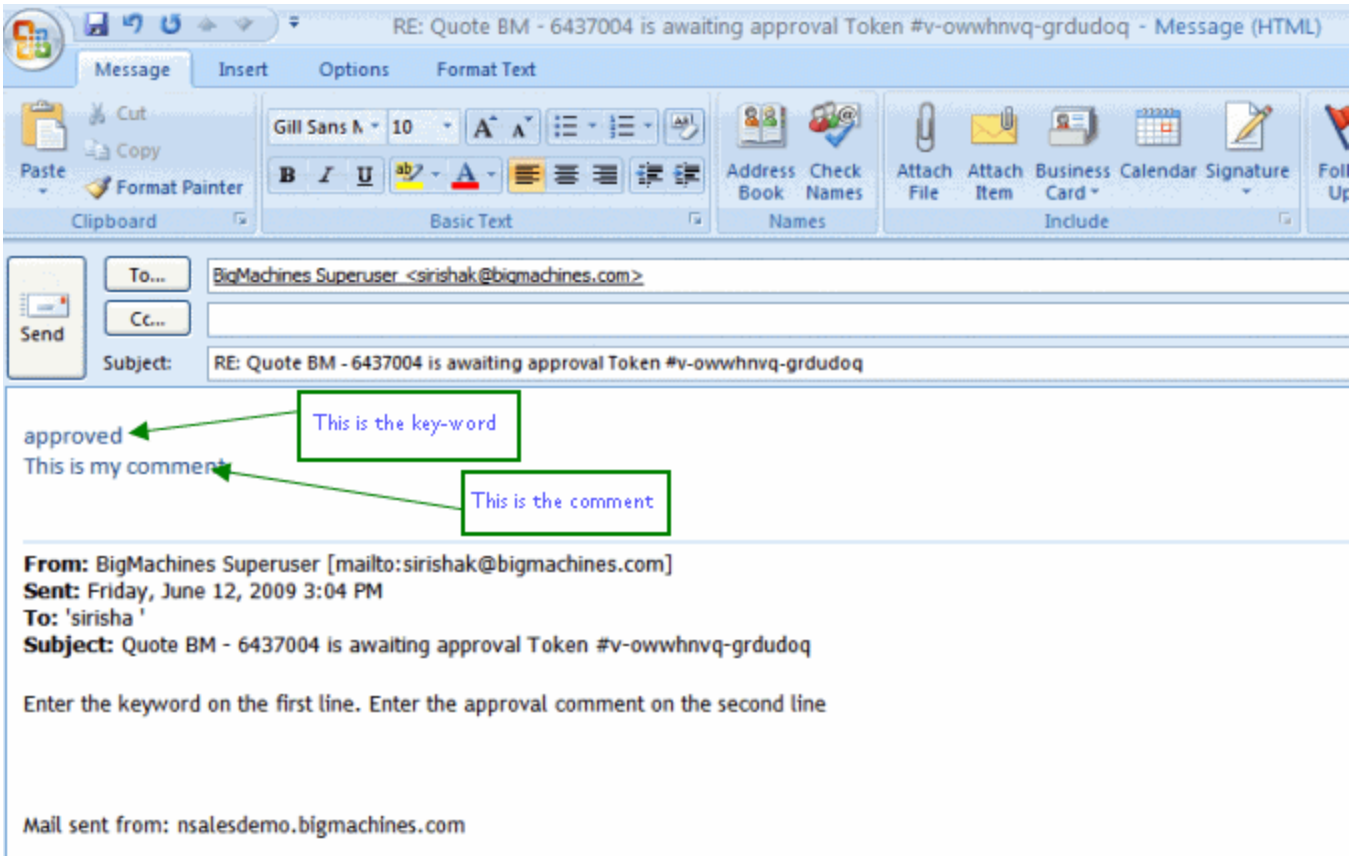
Sample Email with Unique Key



The approver will need to reply to the notification email to perform the modify action. **The subject line of the email must contain the unique key**.

- The first string on the first line of the email will be the keyword associated with the action that the user is trying to perform.
- The second line of the email can be the comment. This comment can be mapped to any text area or history type attribute in commerce.

Sample Reply to the Notification Email



Then, BigMachines sends an email indicating if the action specified by the user was performed successfully or if an error was encountered.

ADMINISTRATION

Setting up Mobile Approvals

1. Navigate to a **modify-type** action. For example, "Approve".
2. Define the **Email Notification keyword** for the action.

   Keywords are not case-sensitive.

3. Multiple keywords can be associated with an action. All possible values can be separated with a tilde in the Email Notification Keyword field on the Action Editor.

- A keyword has to be unique.
- A keyword can have alphabets, numbers and underscore.
- A keyword cannot have multi-byte characters
- A keyword can be associated with only one action within a process

4. Specify a text area or history type attribute in the Approval Comment Mapping field in the Action Editor.
5. The comment specified by a user in the email will be mapped to the specified attribute.
6. On a Transition Rule,set up the Mobile Devices Notification. Select the option Allow response to Trigger an Action.
7. When this option is selected, BigMachines will automatically generate the unique key in the subject line of the outgoing email.

NOTES



The user specified comment can contain multi-byte characters.



BigMachines validates that the user email is registered with the system before sending the unique key. Therefore, only a user with a valid BigMachines account can receive the unique key and can perform the modify action via email. Inactivated users will not be sent a key.

The user responding to the email is used for permissions on actions and any CRM integrations.



The person replying to an approval email must have the same email address as the person to whom the approval notification was sent.

RELATED TOPICS

DEFINE NOTIFICATIONS

OVERVIEW

Here you can manage any notifications that are sent when a transition rule triggers.

ADMINISTRATION

Follow these steps:

1. Navigate to a Transition Rule.
2. Click the **Define Notifications** button.
3. Select the Method(s) of Notification. You can check either or both the options provided:
    1. **Send XML.**
    2. **Send Email**. *Checking this option displays the Email Information section where you must fill out the following information:*

Header	Options
Sender	**Simple Sender Email Address:** Enter comma-delimited email addressed in the text field. **Advanced Sender Email Address:** Click Define Function to write a BML script that will populate the sender email address(es).
Recipient(s)	**Simple Recipient Email Address:** Enter comma-delimited email addressed in the text field. **Advanced Sender Email Address:** Click Define Function to write a BML script that will populate the recipient email address (es).
Cc Recipient (s)	Select additional recipients to be copied. Choose from: • **Document Creator**: Person who created the previous main document in a process containing multiple main documents • **Transaction Creator:** Person who created the transaction. • **Next Step Participants:** Person(s) who belong to the next step in the workflow. • **Current Step Participants:** Person(s) who belong to the current step in the workflow.
Subject	Type text, up to 255 characters, for the Subject line of the notification. You can also append current document attributes, on which the action is present, which will be dynamically populated when a notification email is sent.
Body	Choose the source of the Body: • **Transaction formatted with an XSL File**: This will output HTML content using a Printer-Friendly XSL/History XSL. • **Body text with Smart Tags**: Type custom text, up to 4000 characters, for the Body of the notification. You can also append current document attributes, on which the action is present, which will be dynamically populated when a notification email is sent.
Attachments	Select attachments to be sent with notification email. • **Transaction formatted with an XSL file:** Use an XSL template from the Printer Friendly and History XSL views. • **Attachment name with Smart Tags:** Name your attachments using the current document attribute smart tags.

RELATED TOPICS

QUOTE STATUS IS INCORRECT

OVERVIEW

It's important when dealing with work flow issues to recognize that there is a distinction between Quote Status and Quote Step.

- **Quote Status** is usually a descriptive attribute on the quote that helps identify what stage of the work flow the quote is in.
- **Quote Step** is a special attribute within BigMachines that is used to determine permissions and document views, send email notifications, and link with integrated external systems.

From time to time, the Status field and Step may get out of Sync, due to changes in logic to one or the other. When this occurs, it is helpful to find the Step ID.

ADMINISTRATION

Understanding incorrect Quote status

1. To find the Step ID, go into your transaction manager and view the quote.
2. Once in the quote, Right Click on the background and select View Page Source. A window will pop up with the source code of the web page.
3. Search for _step_id_value. You'll see something like this: input type=hidden name=_step_id value=12345. The number in the value attribute is the step id, in this case 12345.
4. To match it to a step, navigate to Admin --> Process Definition -->Quick Links. Expand the List of Steps.
5. Maximize the Quick Links window to make it fill the whole screen. Now hover over each step with the mouse, paying attention to the link displayed at the bottom of the screen; something like: javascript:openPage('workflow/edit_step.jsp?process_id=9999999;id=12345');

The number at the end of the link - in this case 12345 again, will match on the step that the quote is currently in. You can drill into this step now to view the User Access Rights, Transition Rules, Email Notifications, and Document Views.

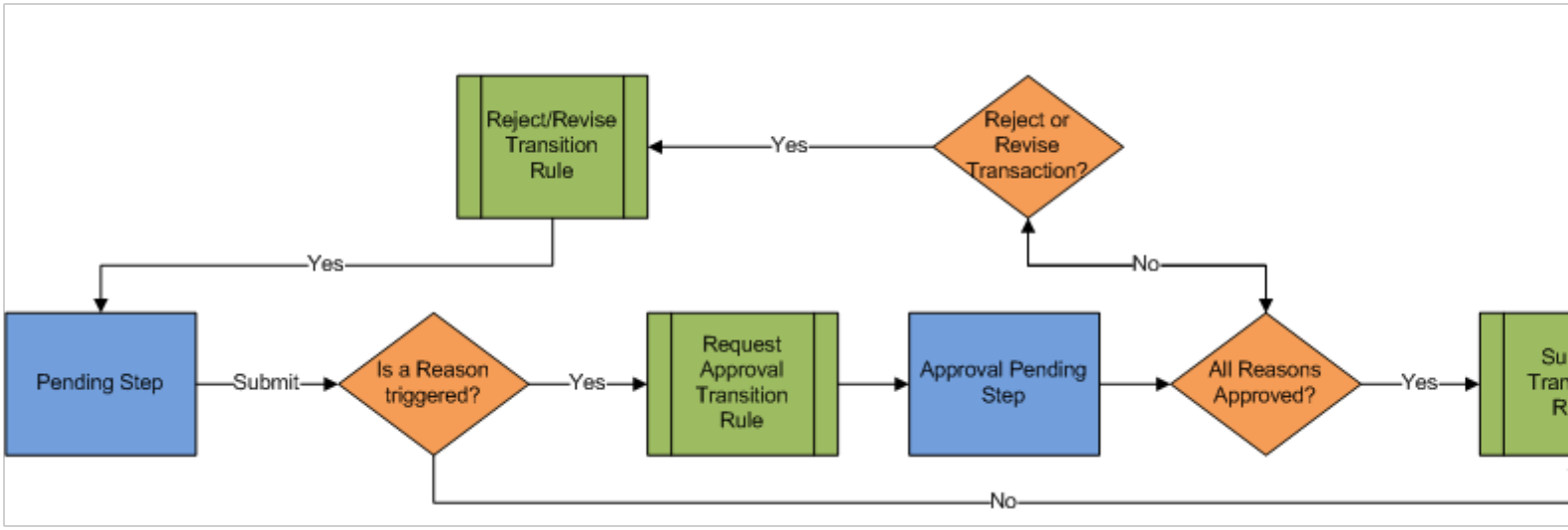
RELATED TOPICS

APPROVAL SEQUENCES OVERVIEW

OVERVIEW

Almost all commerce workflows involve multiple approval stages. As an example, let's say one of your sales agent submits a quote, perhaps it has to be approved by multiple Sales Managers. Additionally, the quote might have to be approved by a Technical Manager before it is finalized. It may also have to be approved by VP if the sale is over $100,000 or has a high discount. At each approval stage, a quote must be approved for one or many reasons. In previous versions of BigMachines, setting up multiple levels of approvals required the use of advanced BML scripts, multiple transition rules, as well as additional workflow steps. The purpose of the Standard Approval is to provide ease of Administration and improved User-Side functionality.

An Approval sequence is triggered when the user submits a quote by clicking the Submit button.. The fullaccessuser can implement a string of Reasons and their corresponding approvers to set up a multi-tiered approval process. If there is no reason for an approver to approve the quote, the submit action is successful and the quote is approved. If a Reason is satisfied, the appropriate approver(s) must Approve or Reject the quote. When the final quote has been revised and the final Approval is performed, the quote is approved. A submitted quote can also be modified using the Revise action.

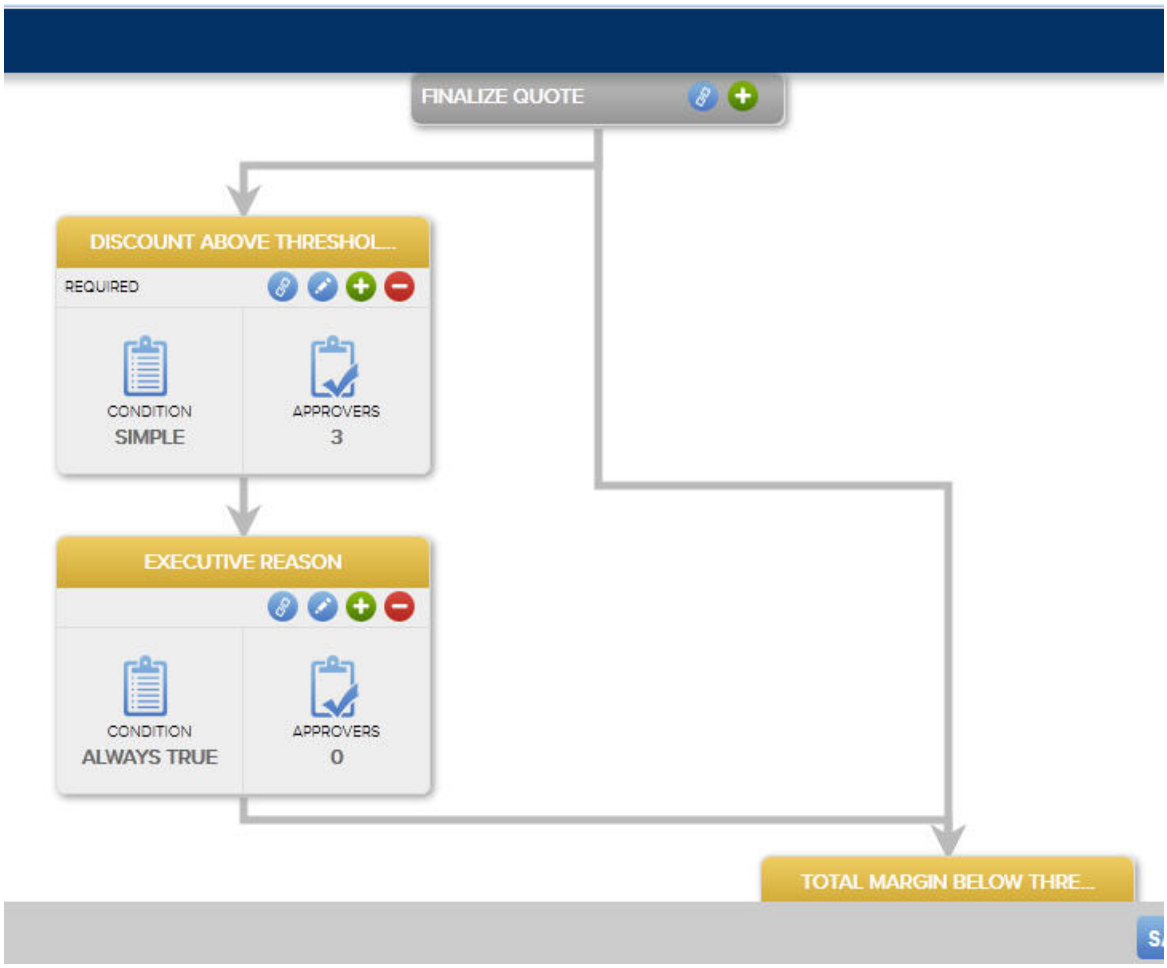


ADMINISTRATION

To use approval sequences, a new Submit Action Set has been created. Once your action set has been created, you can add reasons that a quote needs to be approved. For every reason, the quote may follow a serial or parallel approval sequence (or both) with one or many approvers. You can set up one or many approval sequences within a commerce workflow using an intuitive and visual editor.

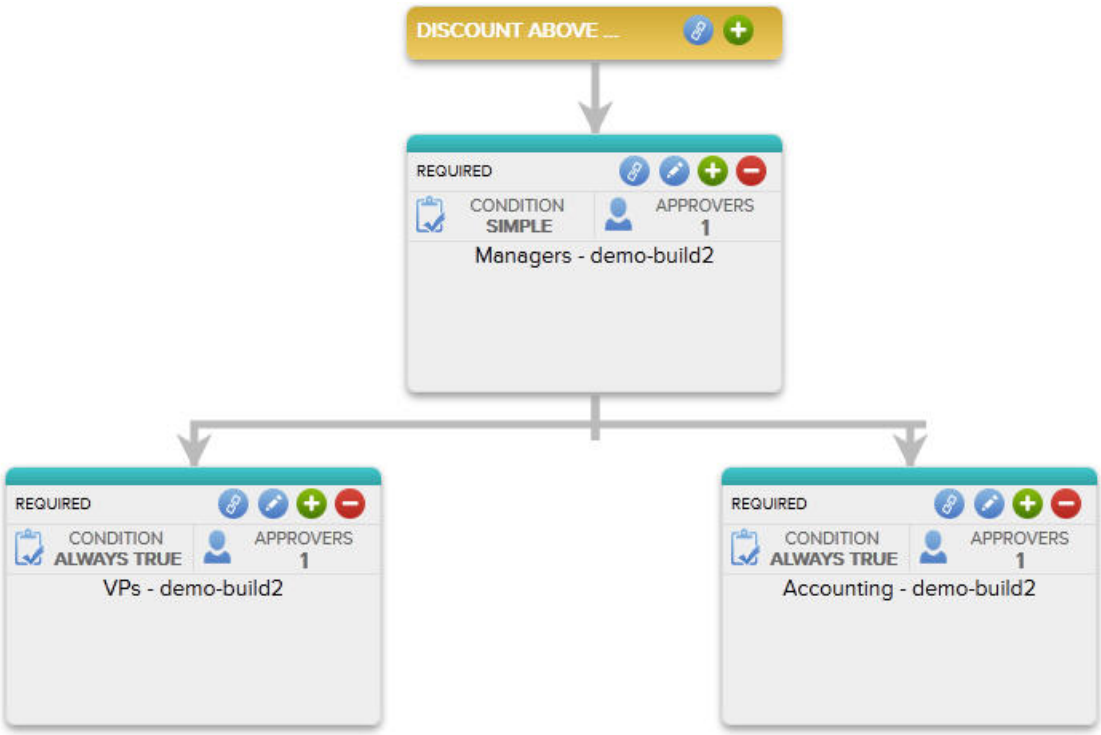
Example of an Approval Sequence

The Submit Action will be listed at the top of the sequence. The graph will then represent serial and parallel reason and approver sequences.



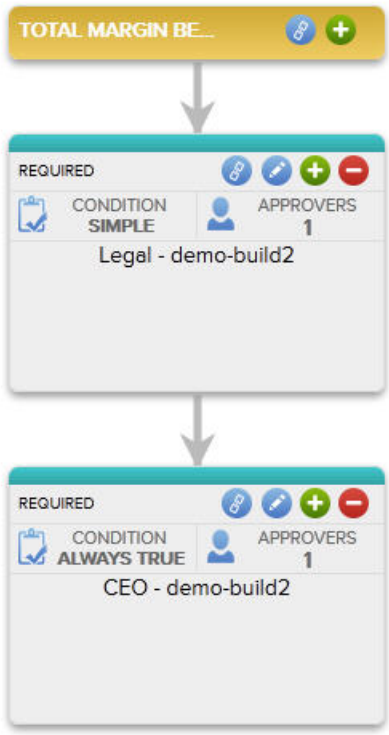
Parallel Approver Sequence

In a Parallel Approver Sequence, both approvers have to approve the quote in order for it to be considered "approved" and move to the next Step in the workflow. Notice in the image below that after the Managers approve, either VPs or Accounting will be required to approve the quote.



Serial Approver Sequence

In a Serial Approver Sequence, the Reason must be approved by a first-level approver BEFORE moving on to the next approver.



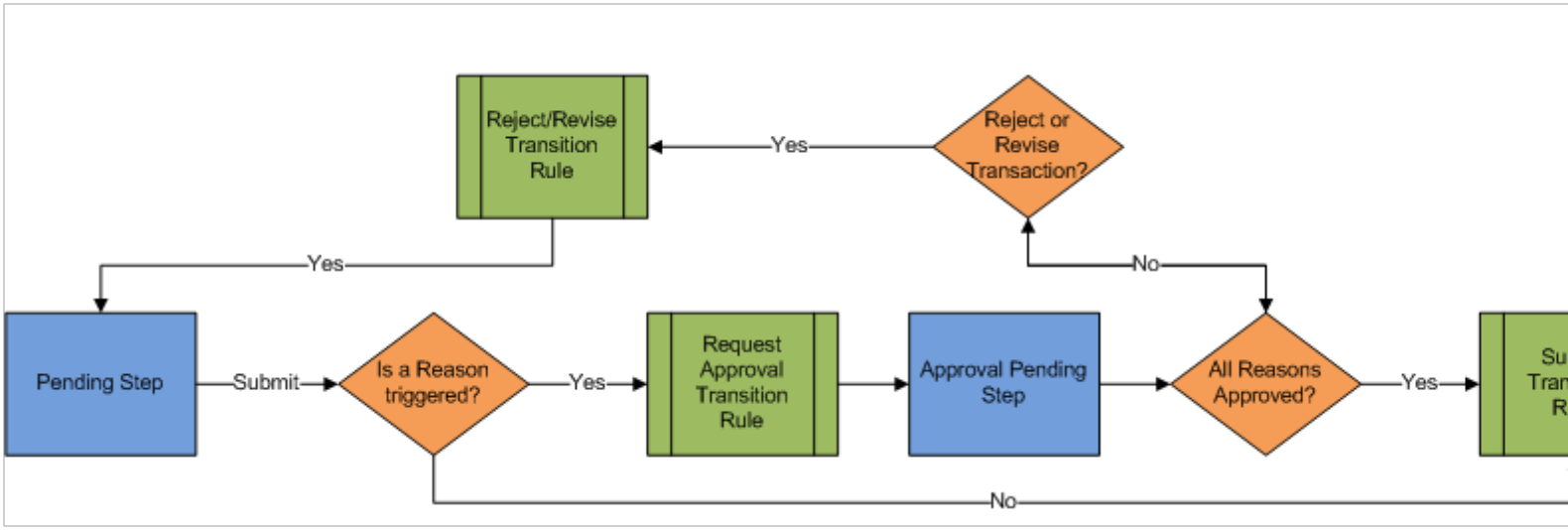
RELATED TOPICS

APPROVAL SEQUENCES OVERVIEW

OVERVIEW

Almost all commerce workflows involve multiple approval stages. As an example, let's say one of your sales agent submits a quote, perhaps it has to be approved by multiple Sales Managers. Additionally, the quote might have to be approved by a Technical Manager before it is finalized. It may also have to be approved by VP if the sale is over $100,000 or has a high discount. At each approval stage, a quote must be approved for one or many reasons. In previous versions of BigMachines, setting up multiple levels of approvals required the use of advanced BML scripts, multiple transition rules, as well as additional workflow steps. The purpose of the Standard Approval is to provide ease of Administration and improved User-Side functionality.

An Approval sequence is triggered when the user submits a quote by clicking the Submit button.. The fullaccessuser can implement a string of Reasons and their corresponding approvers to set up a multi-tiered approval process. If there is no reason for an approver to approve the quote, the submit action is successful and the quote is approved. If a Reason is satisfied, the appropriate approver(s) must Approve or Reject the quote. When the final quote has been revised and the final Approval is performed, the quote is approved. A submitted quote can also be modified using the Revise action.

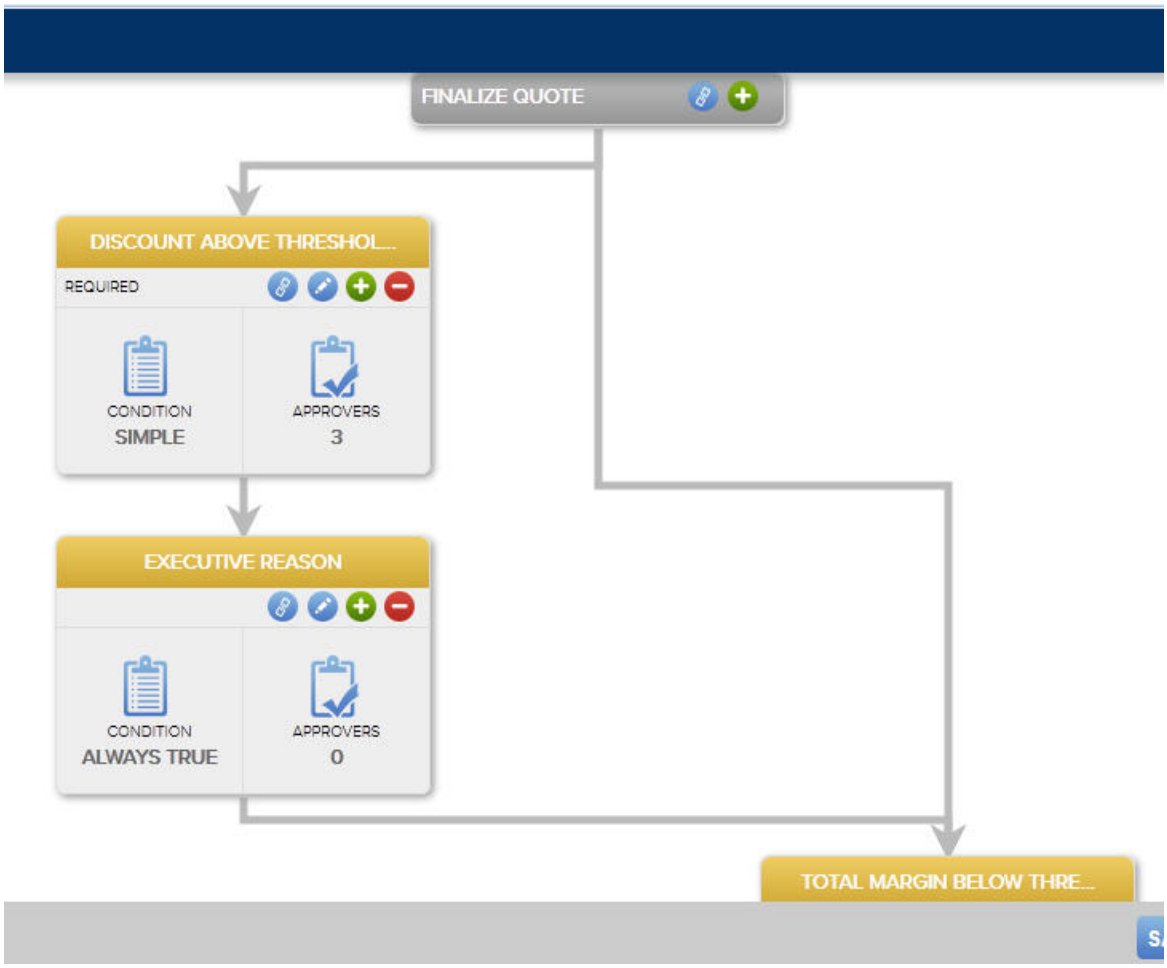


ADMINISTRATION

To use approval sequences, a new Submit Action Set has been created. Once your action set has been created, you can add reasons that a quote needs to be approved. For every reason, the quote may follow a serial or parallel approval sequence (or both) with one or many approvers. You can set up one or many approval sequences within a commerce workflow using an intuitive and visual editor.

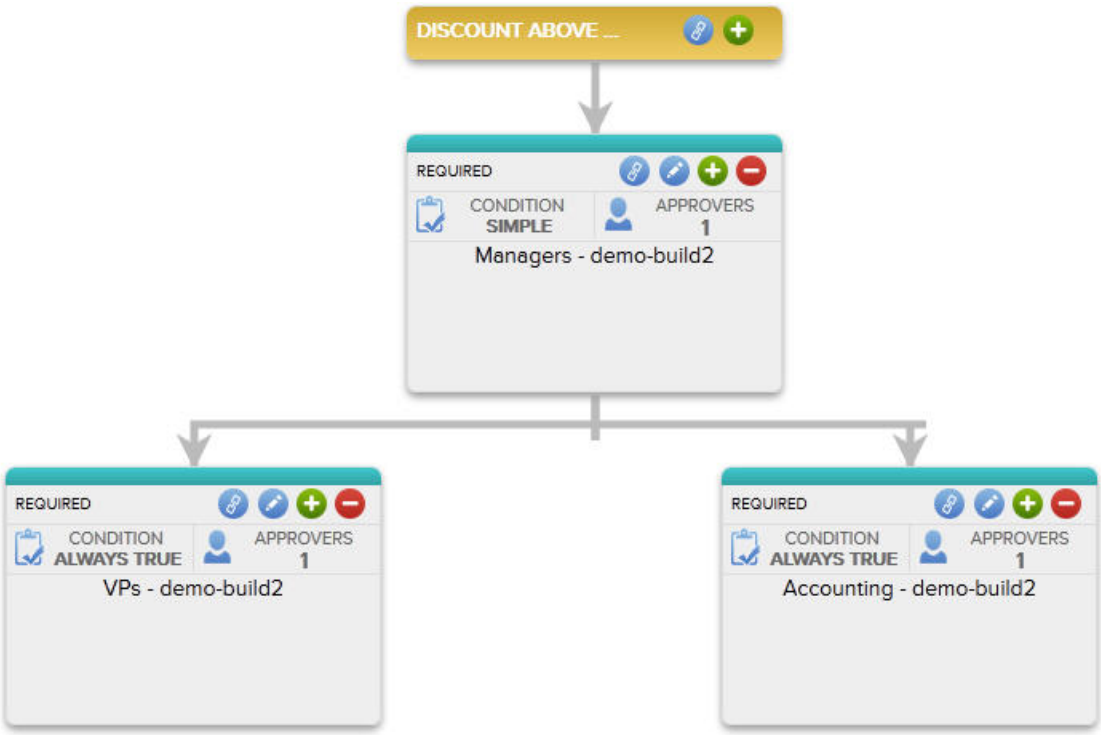
Example of an Approval Sequence

The Submit Action will be listed at the top of the sequence. The graph will then represent serial and parallel reason and approver sequences.



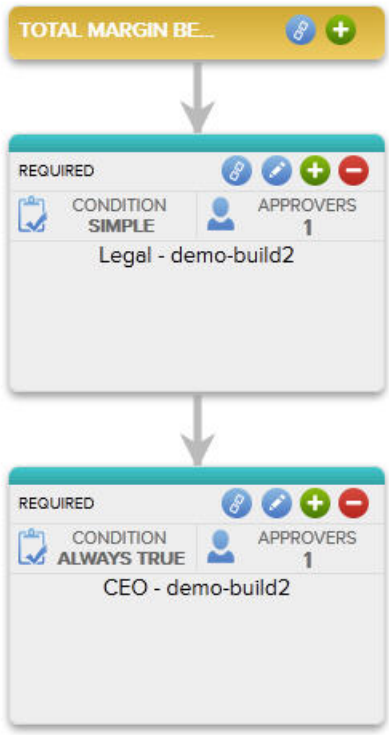
Parallel Approver Sequence

In a Parallel Approver Sequence, both approvers have to approve the quote in order for it to be considered "approved" and move to the next Step in the workflow. Notice in the image below that after the Managers approve, either VPs or Accounting will be required to approve the quote.



Serial Approver Sequence

In a Serial Approver Sequence, the Reason must be approved by a first-level approver BEFORE moving on to the next approver.



RELATED TOPICS

MY APPROVAL SEARCH

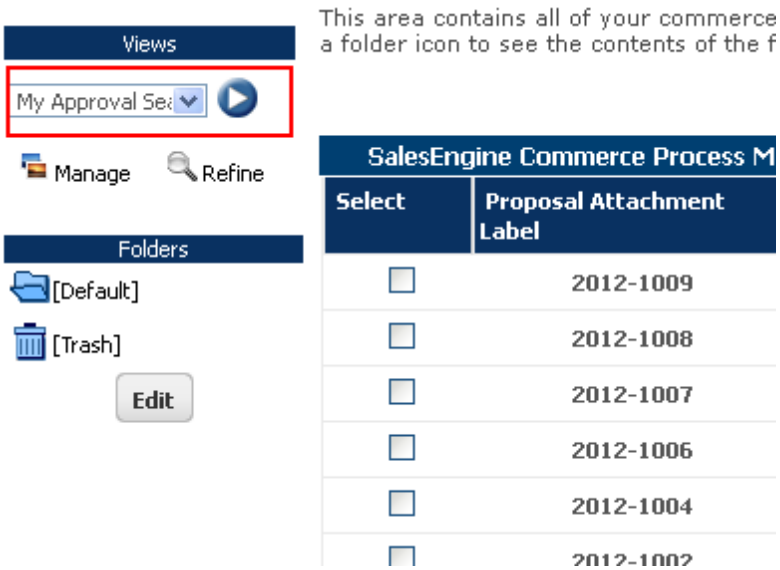
OVERVIEW

BigMachines allows users to search for pending approvals through the View Manager. Users have pending approvals waiting in a queue when there are quotes they need to approve. This makes it easy to enter each individual quote to approve or reject them.

ADMINISTRATION

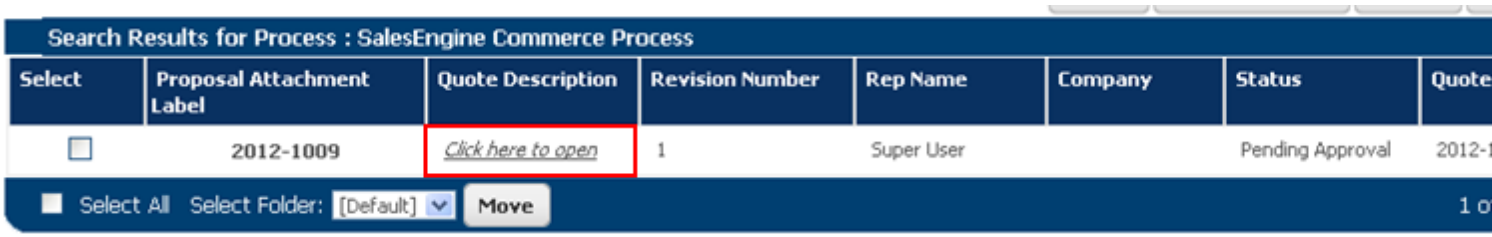
Viewing Pending Approvals

1. Navigate to the Transaction Manager from the header or side navigation menu.
2. Select **My Approval Search** from the Views drop-down menu and click the **Blue Arrow** to perform the search.

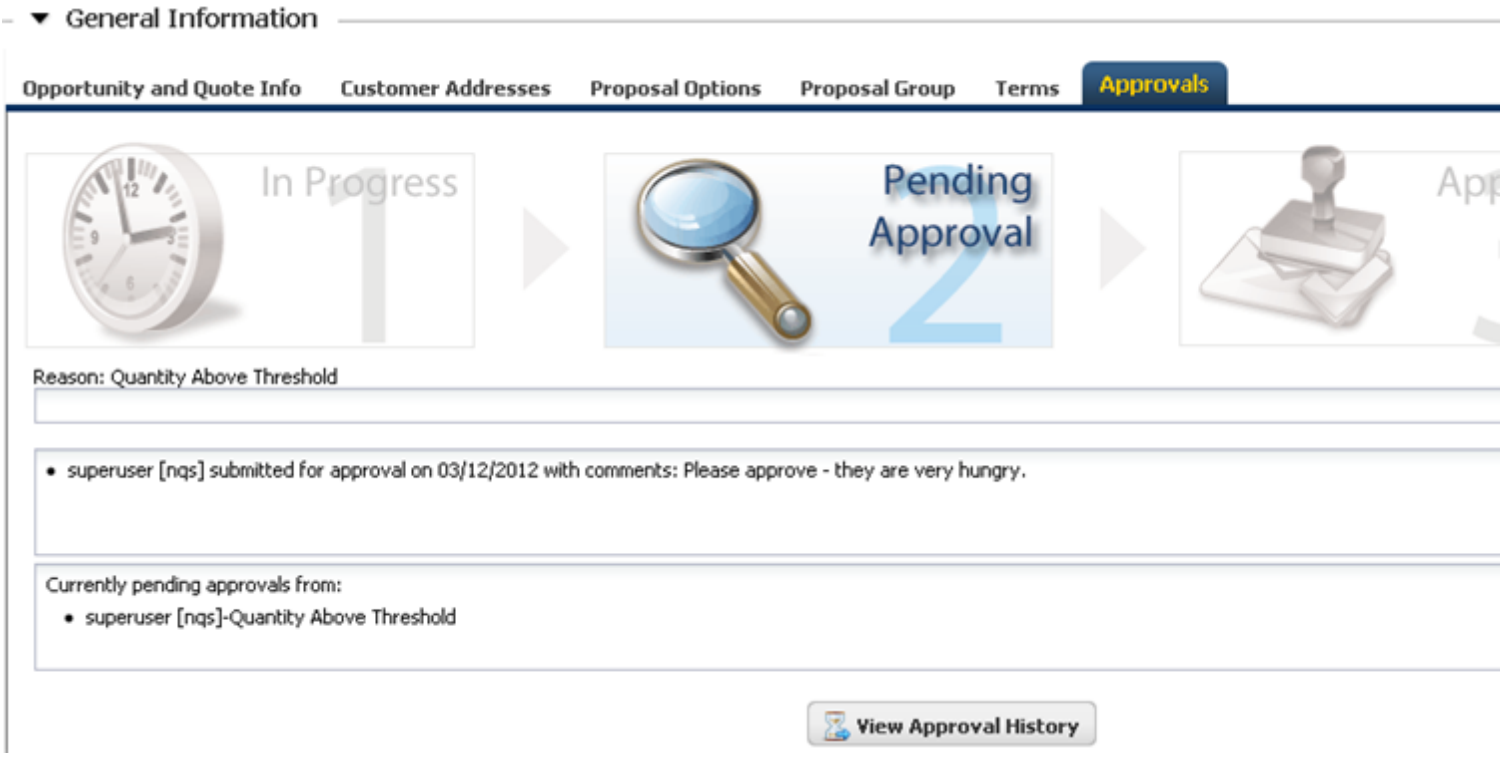


You'll notice a list of your pending approvals in the transaction manager search results page.

3. Click the ***Click here to open*** link under the Quote Description column.This will take you directly to the quote you need to approve.



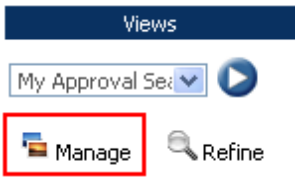
4. Click on the **Approvals tab** on the main-document and either approve or reject the quote.



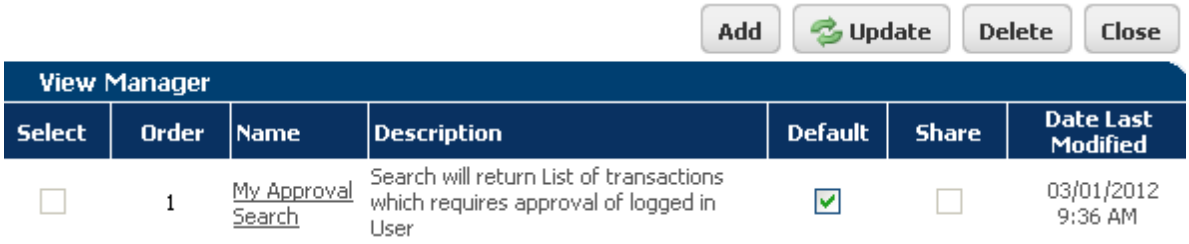
Managing My Approval Search View

You can set My Approval Search as the default search view on the transaction manager. In doing so, when the user clicks on the transaction manager, they will always see their pending approvals first.

1. Navigate to the **Transaction Manager** from the header or side navigation menu.
2. Click **Manage** underneath the Views header.



3. Select the **Default checkbox** to set My Approval Search as the default view.



4. Click on My Approval Search under the Name column to change the name of the search.

RELATED TOPICS

EDIT APPROVALS

OVERVIEW

Approval Sequences manage the reasons a quote should be approved and who is responsible for approving the quotes. In this article, we'll show you to how create and manage reasons within an approval sequence.

ADMINISTRATION

Managing Reasons

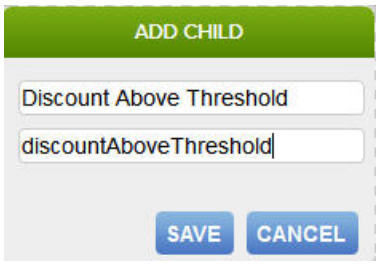
1. Navigate from Admin Home Page > Process Definition Quick Links > Select a Process > Quote > List of Actions > Submit [type] Action
2. Click **Edit Approvers**.



3. Click the **Add Child** icon on the submit action header.

> If this is the first reason in a new sequence, the Add Child object will pop-up automatically.

4. Enter a **name and variable name** on the Add Child pop-up , then **click Save**.

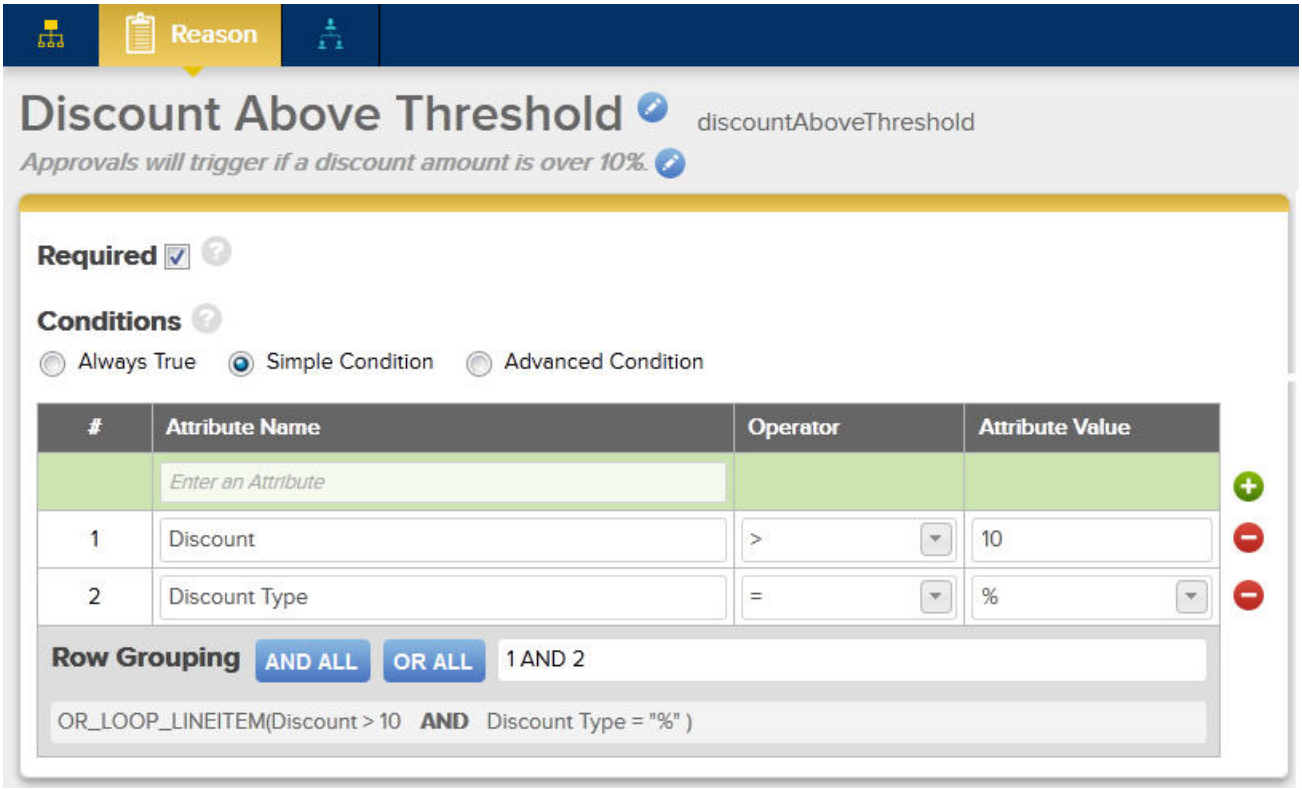




5. Click the **Edit icon** on the reason to edit the name, description and set the reason conditions.
6. Select the **Required option**, if desired.

   When selected, reasons that follow this one in the sequence will be skipped if this reason condition evaluates to false.

7. Enter **Conditions** for the reason. *Conditions are evaluated when the user clicks Submit.*
   - If the conditions are returned true, then the approval sequence for the reason is executed and the system will modify the Approval Status and Reject/Approve icon attributes.
   - The condition editor is similar to that in Configuration except:
     - The condition attributes that can be selected will be commerce attributes: System Attributes and Document Attributes.
     - The advance function editor will be the same as the function editor for the commerce library functions. See Function Wizard for more details.



8. Click **Save.**

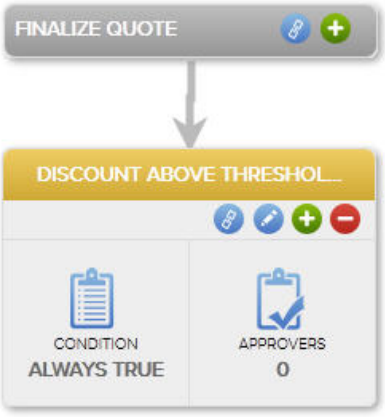
Managing Approvers

Now that you have created your reasons, you can add approvers to the sequence. REMEMBER:

- **Parallel Approver Sequences:** both approvers have to approve the quote in order for it to be considered "approved" and move to the next Step in the Workflow.
- **Serial Approval Sequences:** In a Serial Approver Sequence, the Reason must be approved by a first-level approver BEFORE moving on to the next approver.

Follow these steps:

1. Click **Apporvers** from within a Reason.

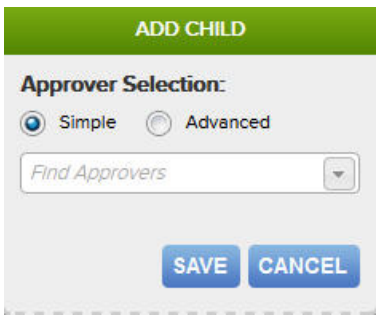


2. Choose an **Approver Selection**: Simple or Advanced.

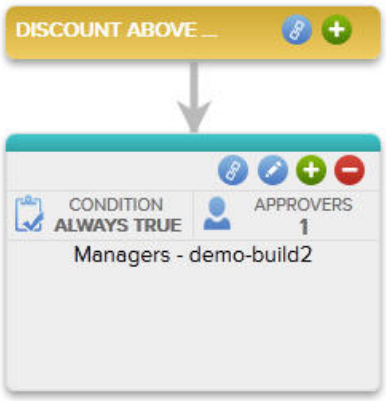
> Advanced approvers functionality requires that at least one approver is returned. Format is "1~ user-~company|2~group~company|":
>
> -- The 1 denotes that it is a user that is being designated as the approver.
>
> -- The 2 denotes that it s a group that is being designated as the approver.
>
> -- Only add the | delimiter if there are multiple approvers.

3. Select **users or groups** from the drop-down menu, then click Save.

> Here you can select groups and/or individual users. If a group is added, only one person in the group needs to approve the quote to move it to the next approval in the workflow. If more than one group or user is selected, then the values will be delimited with an "or".



4. Click the Edit icon on the **Approver** to enter conditions and manage notifications.



5. Select the **Required option**, if desired.

> When selected, approvers that follow this one in the sequence will be skipped if this approver condition evaluates to false.

6. Select the Notification type:
   - **None:** Select to disable all email notifications (ie. Pending Approval, Retraction, Rejection, etc.).
   - **Template:** Select from the pending approval notifications that were created from an Email Template. *If you select Template, you'll see two more options appear: Email Template and Attachment.*
   - **Advanced:** Use BML to create the pending approval notification.
7. Select the **Email Template** to use for the pending approval notification email.

> This is only available if you selected Template in Step 6.

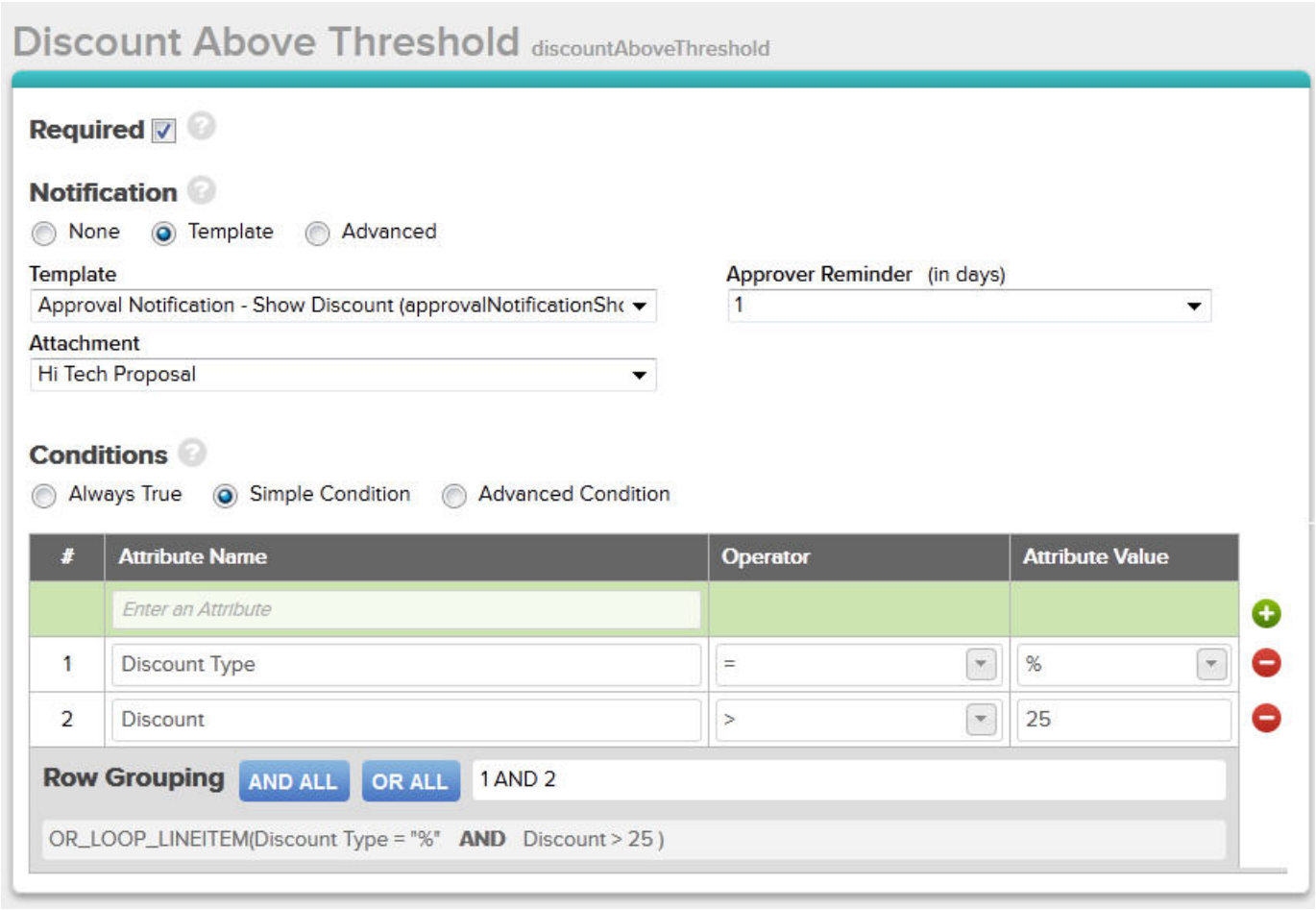
8. Select **ONE attachment** to include in the notification email. This can be any document created by the Document Engine or a Printer-Friendly/History XSL.

> This is only available if you selected Template in Step 6.

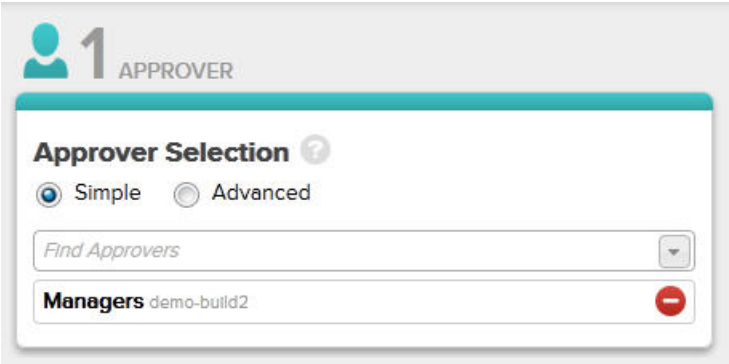
9. Set an **Approver Reminder** (in days). This will send a reminder to the approver, if they have not responded, every x(1-7) amount of days.

10. Enter the **conditions** for when this approve will need to approve the quote. You can select from:

    - **Always True:** The user/group will always to have to approve the quote for the reason.
    - **Simple Condition:** Set conditions, row by row, that must be met for the user/group to approve the quote.
    - **Advanced Condition**: Use BML to set conditions that must be met for the user/group to approve the quote.



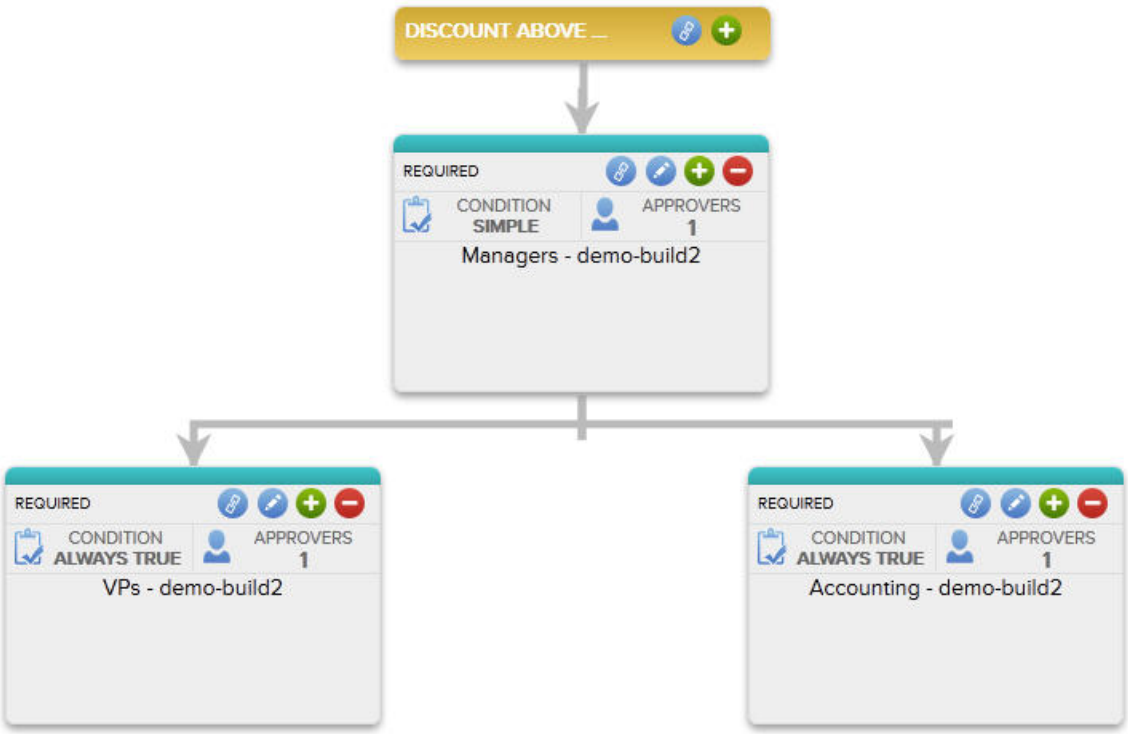
11. Edit the users at any time by following Steps 3-4 on the Approval.



12. Click **Save**.
13. Click the Add Child icon on the Approver to add additional approvers. Follow steps for each approver in the sequence.

Final Result

Once you have added all approvers for a sequence, you will see a visual representation similar to the image below. The Reason will be the header at the top, with the approvers in the graph beneath it:



NOTES



You can drag-and-drop reasons from one parent/child to another.



If more than one reason is coming from a parent, then these are considered to be parallel reason flows. Parent/child relationships are considered serial flows. In serial flows, once a parent reason has been approved, a child reason notifications are sent.

Email templates can only be used for pending approval notifications.

A reason will only get sent to a delegated approver when the approver is defined as an individual, not as a member of a group.



Only one XSL view or document can be attached to an email notification.

If you delete a parent reason or approver, it will also delete the child.

RELATED TOPICS

RULE INPUTS AND FUNCTIONS

OVERVIEW

Rule Inputs

Approval History

This attribute lists all approvals done for a transaction since the last time the approval sequence was reset. In BML, Approval History is returned as a 2-D array, where each row provides the history for each reason that has been either fully or partially approved or rejected. Each history detail includes the approver, the company of the approver and the action that he performed in the following format:

```
userLogin~companyName~Approve //if user approved
```

OR

```
userLogin~companyName~Reject //if user rejected
```

For Example:

If the Pricing Reason has been approved by a manager and Technical Reason has been approved by an engineer and rejected by a product manager, the approval history in BML is:

pricingReason	manager~Acme~Approve	
technicalReason	engineer~Acme~Approver	productManager~Acme~Approve

---

Approval Status

This attribute lists all the currently pending approvers for a transaction. In BML, Approval Status is returned as a 2-D array, where each row provides status for each active reason. Each pending approver detail includes the approver login or group name, the approver or group company name in the following format:

```
userLogin~U~companyName //when the approver is an individual user
```

OR

```
groupVarName~G~companyName //when an approver is a group
```

For Example:

Say an approval is pending for the Pricing Reason from the manager and for the Technical Reason from both the product manager and either the engineering group or the architect. The Approval Status is a 2-D Array:

pricingReason	manager~U~Acme	
technicalReason	engineer~G~Acme|:|architect~U~Acme	productManager~U~Acme

---

Additional Rule Inputs

Rule Input	Description
Submitted_by	This attribute records the user that initially performed the Submit action.
Submit_comment	Submit comment can be used in a notification rule to provide the comment given by the submitter.
Approve_reject_comment	This can be used in a notification rule to provide the comment given by an approver when approving or rejecting a reason.
Curr_reason_name	This can be used in a notification rule to provide the current reason name.
Curr_reason_variable_name	This can be used in a notification rule to provide the current variable name.
Curr_reason_description	This can be used in a notification rule to provide the current rule description.
Curr_reason_approver	This attribute can be used in a notification rule to provide the login of the approver that has just approved.
Next_approver	This attribute can be used in a notification rule to provide the name of the user to whom the notification is being sent.
Document_link	This can be used in a notification rule to provide a link to the transaction document that needs approval.

Functions

The getreasonstatus() function evaluates whether a reason has been completely approved or not. The syntax for this function is:

```
Integer getreasonstatus(String Curr_reason_variable_name)
```

Return Value Format: Integer Constraints.

Attribute	Description
BM_REASON_STATUS_INVALID	No reason exists in reason tree with given variable name
BM_REASON_STATUS_INACTIVE	Reason exists in reason tree, but the condition for the reason is not met.
BM_REASON_STATUS_PENDING	Active reason has not been completely approved.
BM_REASON_STATUS_APPROVED	Reason that has been completely approved.
BM_REASON_STATUS_REJECTED	Reason that has been completely rejected.

NOTES



Approval attributes, like submit comments, approve/reject comment, document link, current approver, next approver, current reason name, and current reason description cannot be populated in a text or text area type attribute. These attribute are intended for use in the approval notifications only and are populated dynamically when used in the advanced notifications.

RELATED TOPICS

SUBMIT ACTION SET

OVERVIEW

The Submit action set is a series of modify-type actions in commerce that are essentially sub-actions of Submit. It is comprised of Submit, Request Approval, Approve, Reject, and Revise.



Submit	submit	Submit	
Request Approval	request_approval_submit	Request For Approval	This action is performed when submit action is clicked and there is at least one reason to approve.
Reject	reject_submit	Reject	This action is performed when a reason is rejected by the approver.
Approve	approve_submit	Approve	This action is performed when a reason is approved by the approver.
Revise	revise_submit	Revise	This action is performed when Revise button is clicked.

One critical point that is often misunderstood is that when the Submit button is clicked, the Submit action and its associated modifications do not necessarily run.

- First, the quote passes through the approval tree defined on the Submit action, and if any of them return true then the Request Approval action will be performed first. Approvers will receive notifications and the Approve and Reject actions will become available to them until they have performed either one, at which time the actions will no longer be visible to them.
- Only when the quote has received its final approval, or if no approvals are required, will the Submit action and its associated modifications actually fire. Plan your status modifications, step transitions, and integrations accordingly!

Sub-Action Set Behavior

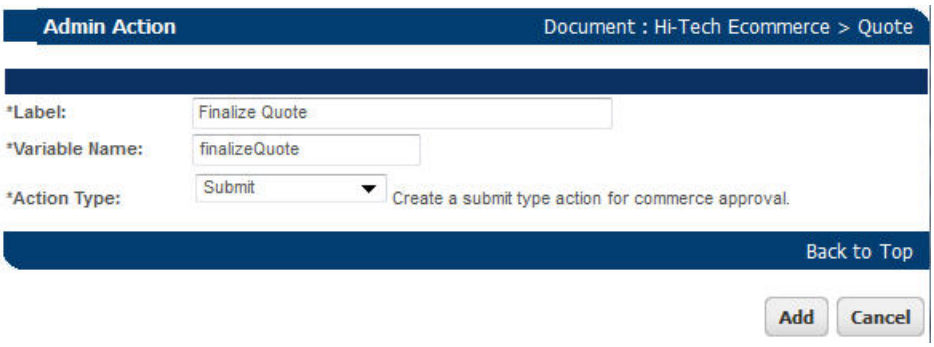
Action or Sub-Action	What Does it Do?	Notes
Submit	The Submit action decides if the quote needs to enter the approval sequence or can be automatically approved. Other times it runs each modify action.	1. The fullaccessuser creates this action as a Main Document Submit type action. 2. Click Edit Approvals to define a hierarchical list of reasons why a quote might have to be approved. Each of the reason will have an approver assigned to it. 3. This action runs when a user clicks the Submit action button in a quote. 4. The admin typically defines a transition rule on this action to take the quote to the Approved step. If there are reasons for which this quote needs to be approved, the Request Approval sub-action will run. 5. The order of operations for a Submit action type is the same as that of a standard Modify action. At the end of the Advanced Validation, the reason rules are run to check if the quote needs any approval.
Request Approval	The Request for Approval action moves the quote into an approval pathway. It is automatically performed when the user clicks on a Submit action and there is a Reason to approve the quote.	1. This action cannot be placed on a commerce document. 2. Auto-Submit option: If the Auto-Submit option is enabled, then all quotes must go through the approval process and the box doesn't show up on the user-side. 3. You will typically define a transition rule to take the transaction to the Submitted for Approval step. If there are no reasons for which this quote needs to be approved, then the Submit action will run. 4. No email notifications should need to be sent out by using the transition rule on this action.
Approve	This action is performed when the user clicks the Approve button.	1. You will typically defines a transition rule on this action to take the quote to an Approve step.

Action or Sub-Action	What Does it Do?	Notes
		2. Transition rules must be defined on this action for step change to an Approve step.
Reject	This action is performed when any Reason for an approval sequence is rejected by an approver.	1. This action can be placed in a group view and made accessible to a user. 2. This must be active in a step for it to be executed.
Revise	The revise action allows the user to revise a transaction document that has been submitted for approval.	1. You will typically defines a transition rule on this action to take the quote to a Pending step. 2. Transition rules must be defined on this action for step change. 3. See Revise Sub-Action for more details.

ADMINISTRATION

Creating the Submit Action Set

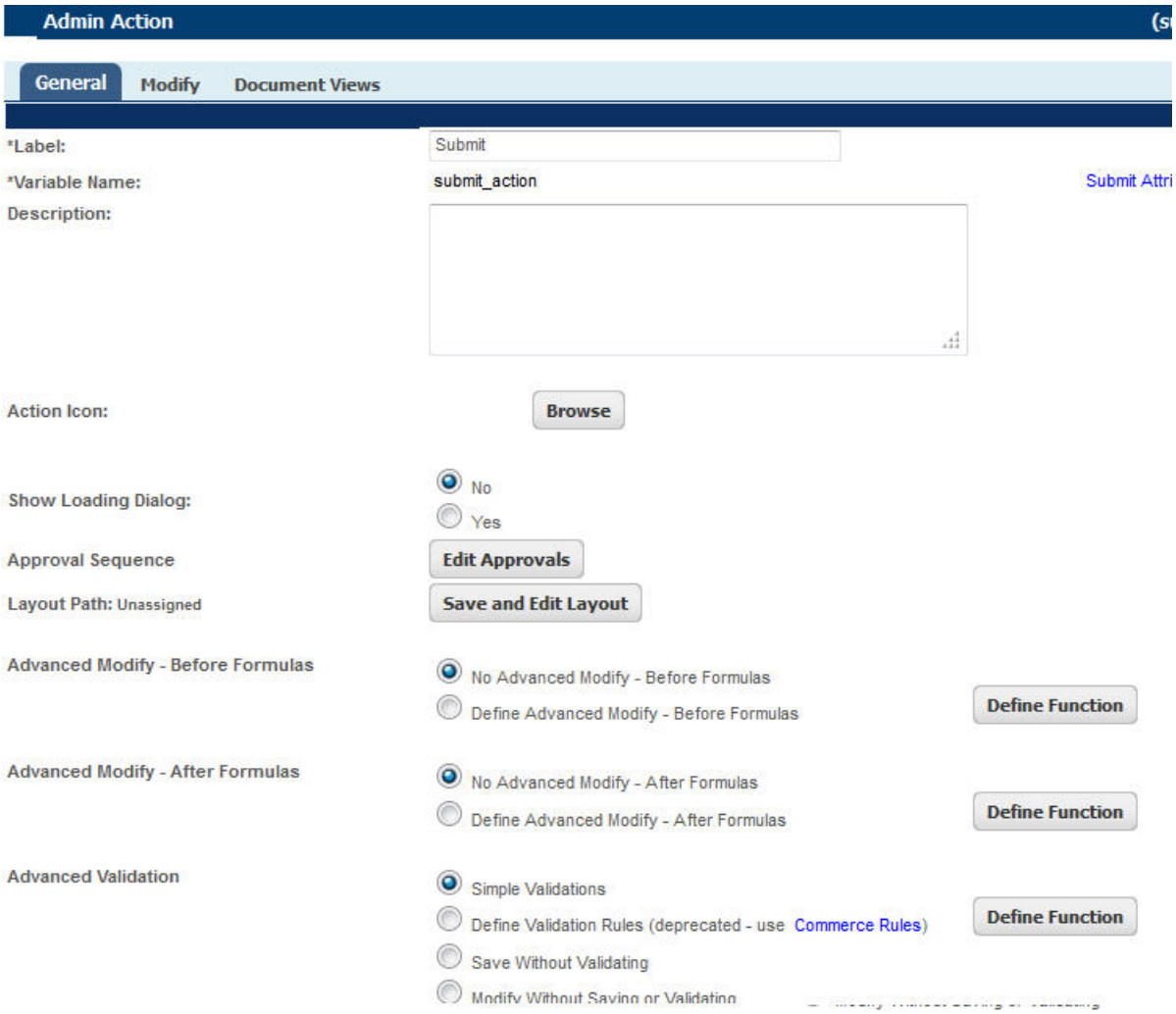
1. Navigate from Admin Home Page > Commerce and Documents > Process Definition > Commerce Process > Documents > List > Quote Level > Actions.
2. Click **Add**.
3. Enter a **Label and Variable Name**.
4. Choose **Submit** as the Action Type.
5. Click **Add**.



Once you have returned to the Actions List page, you should see your new Submit actions along with 4 sub-actions: Request Approval, Reject, Revise and Approve. These can each be edited.

6. Edit the submit action as you would any type of modify action.

7. Click the **Edit Approval**s button to create reasons and approvers within the Approval Sequence.



If you delete the Submit action is deleted, it will also delete all associated actions and attributes.

Revising a Sub-Action (Reset Reasons)

Resetting of reasons is typically used when a Quote is required to step through the entire Approval Sequence, or portions of the sequence, even if some of the Approvals have already been received. An approval reason has 4 statuses: Approved, Rejected, Pending or Ignored.

Resetting & Approval Statuses

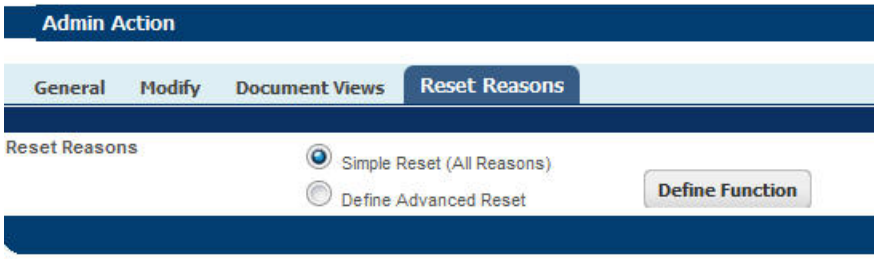
- **Resetting a Reason that is in Approved or Rejected:** The reason was triggered and already received an approval (or rejection). Resetting the reason means it will be re-evaluated upon a Submit action and if the condition is met, an email

notification will be sent and a new approval/rejection will be required.

- **Resetting a Reason that is in Pending:** The reason was triggered but was neither approved nor rejected. Resetting the reason means that it will be re-evaluated upon a Submit action and if the condition is met, an email notification will be sent and a new approval/rejection will be required.
- **Resetting a Reason that is in Ignored:** The reason was not triggered in the initial request so resetting it has no affect.

---

1. Navigate to the Quote Level Actions > Revise Sub-Action.
2. Click on the **Reset Reasons Tab**.
3. Select one of the following:
    - **Simple Reset (All Reasons):** If the checkbox is selected, the reason will be reset. If the checkbox is not selected, the reason will not be reset. NOTE: The default setting is for all reasons to NOT BE reset.
    - **Advanced Reset:** Write an advanced function that will return the reasons that should be reset. The return format can be an array or delimited string of the reason variable names.



If the desired outcome is that, regardless of changes or revisions, the Quote only needs an Approval once, then the Reason or Reasons should not reset.

The Revise sub-action cannot activate reasons or submit the transaction. Revise will always perform only its own step transition and all active approval nodes will always be deactivated. Approval workflow cannot be resumed until the request for approval action is performed again.

NOTES



When setting transitions, try to use just one pending approval step. Adding multiple steps increases complexity. It is recommended to use a variety of participant profiles instead.

Keep both the parent "Submit" action advanced modification and the sub-action "Submit for approval" advanced modification equivalent.

RELATED TOPICS

SUBMIT ATTRIBUTE SET

OVERVIEW

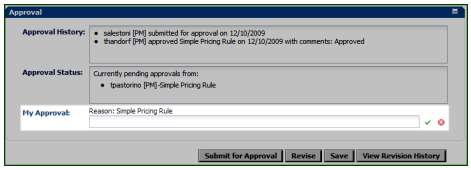
The Submit Attributes Set is generated by the system after the creating of a standard Submit action. The attributes are added to the Quote Attributes List page.

Finalize Quote Submit Attributes	Submit Attribute Set	Submit attributes are created together and removed together. Submit attribute set is created only when Submit type action is created.

When you click on the attribute set name, you can view the details of the individual attributes:

My Approval

A special attribute for the Reject/Approve icons is created automatically when a standard Submit Action is created. The Reject/Approve icon attributes includes a set of approve and reject icons for each reason that has been defined. If reasons are added or modified, the attribute automatically updates. The admin user does not have the ability to modify other properties of the individual icons. The attribute value gets saved to the database when the Approve or Reject button is clicked on the commerce document. When the user clicks the Approve or Reject icon, it updates the value of the reason the user approved or rejected the quote and also the comments that the approver enters. This is done to ensure that two different approvers are not over-riding each other's approvals.



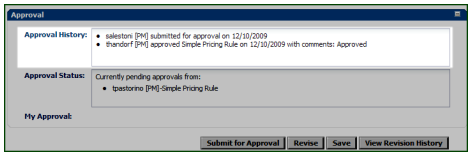
Approval Status

This attribute is created automatically when a standard Submit Action is created. The Approval Status attribute lists all of the approvals that are currently pending for a quote. If the quote is not waiting for approvals on anything, the Approval Status section will read "No Approvals Pending".



Approval History

The Approval History attribute is automatically created when a standard Submit Action is created by the fullaccessuser. The Approval History attribute keeps a record of everything that is happening with the approvals. It shows the person who created the quote, when the quote was submitted for approval and also a history of who has approved the quote and the date of the approval.



Approval Revision

The Approval Revision history attribute is an integer-type attribute that will count the number of times the quote has been revised.

NOTES



The Submit Attributes must be added to the Commerce Document by Using the Layout Editor.



Order of Operations: Sub doc Modify tab of Update Line item > Update Line Item Validation > Modify Tab of Standard Approval action > Advanced Modification of Standard Approval Action > Advanced Validation of Standard Approval action > Recalculate buy side reason approver tree > Step Transition > Save Reason Tree Info/Update Approval History/Updated Approval Status. NOTE: The first two operations are run ONLY when the line item data has changed.

The Modify Tab/Advanced Modification of the Submit action provide the values on which the tree calculation is based for the first time. If there are reasons to approve at the end of the Submit action, the Request for Approval action is automatically run and the tree is calculated based on the values derived from the modify tab/advanced modification of the Request for Approval action. Therefore, modify tab/advanced modification of Submit Actions should be the same as the modify tab/advanced modification of the Request for Approval action, especially for attributes that are used in the tree calculation (attributes that are used in the reason condition or the next approver condition).

Getreasonstatus() is updated in most cases at the end of all the operations, so it will not return the statues of a reason in the same pass. In general, getreasonstatus(except for reject)/approval history are updated at the end of the action so the updated history/status is not available in the same action that changes the status/history.

Reset reasons are removed from approval history. However, approval history for a rejected reason is not reset before the Reset Reason tab is run. So, you can check Approval History for the status of a recently rejected reason in the Reset Reason tab (but remember that the rejected reason will be removed from the history right after Reset Reason).

RELATED TOPICS

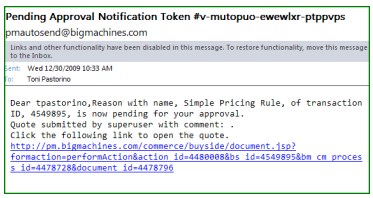
NOTIFICATIONS

OVERVIEW

A major feature of Approval Sequences are the notifications that are sent to the involved parties. Approvers that become active are sent approver notification emails to alert them that they have to approve a quote. When a reject/revise action resets reasons, the pending approvers are automatically sent retraction emails. At the end of the approval process, when the quote is either approved or rejected, the final notification is sent to the submitter. Default notifications are sent automatically by the system. The fullaccessuser can choose to override the email body of some of these notifications by using advanced functions. The notification email advanced function return format allows the fullaccessuser to specify a document engine XSL so the transaction printer-friendly can be attached.

Pending Approval Notifications

This is the standard pending approval notification. You can use Email Templates to create the pending approval notification email.


Pending Approval Notification Token #v-mutopuo-ewewlxr-ptppvps
pmautosend@bigmachines.com
Links and other functionality have been disabled in this message. To restore functionality, move this message to the Inbox.
Sent: Wed 12/30/2009 10:33 AM
To: Toni Pastorino

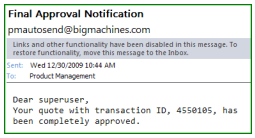
Dear tpastorino,Reason with name, Simple Pricing Rule, of transaction ID, 4549895, is now pending for your approval.
Quote submitted by superuser with comment: .
Click the following link to open the quote.
http://pm.bigmachines.com/commerce/buyside/document.jsp?formaction=performAction&action_id=4480008&bs_id=4549895&bm_cm_process_id=4478728&document_id=4478796

Rejection Notification


Reason Rejection Notification
pmautosend@bigmachines.com
Sent: Wed 12/30/2009 10:36 AM
To: Product Management

Dear superuser,
Reason with name, Simple Pricing Rule, of transaction ID, 4549895, has been rejected by Toni with the comment : Please add one more server and will approve..

Approval Notification


Final Approval Notification
pmautosend@bigmachines.com
Links and other functionality have been disabled in this message. To restore functionality, move this message to the Inbox.
Sent: Wed 12/30/2009 10:44 AM
To: Product Management

Dear superuser,
Your quote with transaction ID, 4550105, has been completely approved.

NOTES



The person replying to an approval email must have the same email address as the person to whom the approval notification was sent.

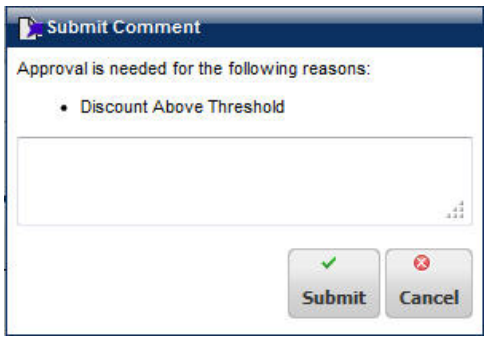
RELATED TOPICS

USER-SIDE FUNCTIONALITY

OVERVIEW

Submitting a Quote for Approval

Assume that a sales representative has just completed a model configuration and has clicked Save. When a user clicks the Submit button on a quote or transaction via a standard approval action, reason rule(s) should run to determine if the quote needs to be approved. If a one or more reasons are triggered, the quote needs to be approved. A Submit Comment dialog opens where the user can enter comments before clicking Submit, thus submitting the quote for approval. The dialog will list all of the reasons a quote might need to be approved.

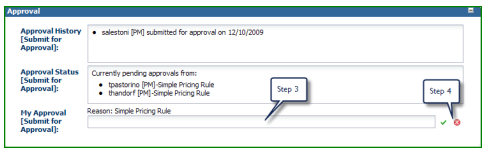


For more information on how to set up reasons and approvers, see the article Edit Approvals.

Approving or Rejecting a Quote

If a user submits a quote for approval, through the standard approvals process, and one or more Reasons are returned as True, then the quote will need to be approved by one more approvers. For more information on setting up approvers, please see the Approvals Tab article.

1. The approver must click on the appropriate Commerce Process Manager.
2. Select the quote pending approval.
3. Leave comments in the text area next to My Approval.
4. Click on the GREEN check mark to Approve and click on the RED "x" to Reject the quote.



Once the quote has been approved, it will move onto any serial approvers or it will remain pending until any parallel approvers have reviewed and either Approved or Rejected it. The Approval History will show who has approved/rejected the quote, any comments left

and the Date it was approved or rejected. Approval Status will show who still needs to approve or reject the quote and Reason for the approval.

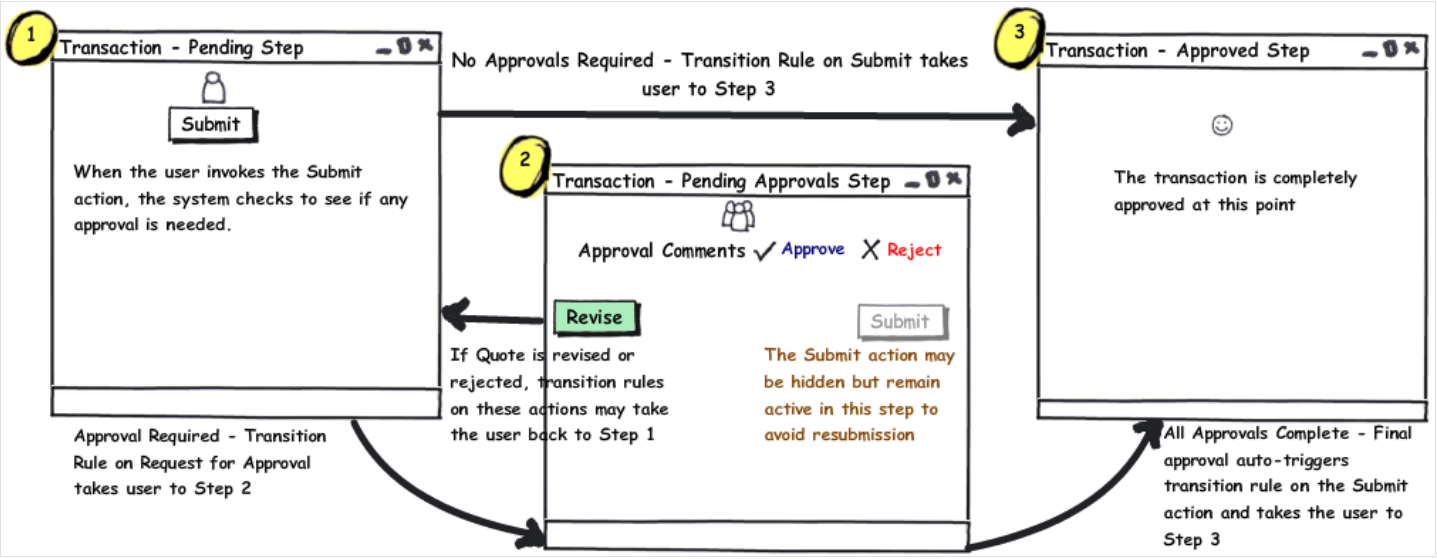
RELATED TOPICS

WORKFLOW INTERACTION

OVERVIEW

In order to use Standard Approvals, the newly created Submit Action and its sub-actions must be added to transition rules within the steps. We can understand how Standard Approvals will interact with the workflow steps by employing an illustration involving the following steps - **Pending**, **Pending Approvals** and **Approved**.

- A quote starts on the **Pending** step. When the user invokes the Submit action, BigMachines checks to see if any approval is needed. If no approvals are needed, the Submit action is performed and the quote is approved. Therefore, on the Submit action, the fullaccessuser must define transition rules to the **Approved** step.
- If approvals are needed, the Request for Approval action is triggered and the quote must be moved to the **Pending Approvals** step. You will need to define a transition rule on the Request for Approval action to take the quote to the **Pending Approvals** step.
- On the **Pending Approvals** step, the fullaccessuser should define a transition rule on the Submit action to the **Approved** step since the final acceptance will automatically trigger the Submit action and quote will be approved.
- You should also define transition rules on the Revise and Reject actions to take the quote back to the **Pending** step.
- The Submit action should be hidden (not inactivated) after the **Pending** step to prevent users from continuously resubmitting. Also, the revise action should only be made visible in the **Pending Approval** step.



RELATED TOPICS

APPROVAL SEQUENCES FAQ'S

Is the Submit action with the reason/approval tree intended to replace steps?

The Submit action is NOT intended to and cannot replace workflow steps. Instead, the submit action works in conjunction with Steps to simplify commerce workflow. The submit action abstracts the reason and approvers tree from the workflow so that the full-accessuser is able to maintain approval sequences independent of Participant Profiles. Besides, the approval sequence is now available to the admin in a graphical format that is easy to maintain. This is useful for the following:

- Decouple approval login and approvers from steps; thereby reducing the number of steps needed in the workflow
- Decouples notifications sent to approvers from notifications sent during step transitions; thereby reducing the number of transition rules
- Makes workflow easier to manage and maintain
- Provides a graphical tool for approval management

Steps and the submit action perform related but different functions. While the Submit action determines the reasons for approvals and the approvers for each reason, steps provide document views.

- Use Steps to:
    - Provide Document View by Participant Profile
    - Define Transition Rules to change the document status
- Use Submit Action to:
    - Define reasons for approval along with the logic that triggers reasons
    - Define list of approvers for each reason along with the logic that trigger approvers
    - Control notifications sent to approvers

What action is performed when the user clicks on Submit? What action is performed when the approver clicks on the Approve/Reject Icon?

- **Submit action:** This action is performed in two different ways:
    - The submitter clicks on this action and there are no reasons to approver. In this case, the Submit action is performed without the transaction having to go through the approval process.
    - The final approver clicks on the Approve icon and there are no more reasons to approve. In this case, when all approvals are done, then the Submit action is invoked automatically after the final Approve action.
- **Revise Action**: This action is performed when the user clicks on the Revise Action.
- **Request Approval:** This action is performed automatically when the user clicks on the Submit action and there are reasons to approve.

- **Approve:** This action is performed when the approver clicks on the Approve icon of the My Approval attribute.
- **Reject:** This action is performed when the approver clicks on the Reject icon of the My Approval attribute.

With the quote in the Approved step, when the user clicked 'Revise', the quote did not appear_to transition back to the Pending step..

In most cases, this happens because the advanced validation on the 'Revise' action functioned, but the simple modifications and advanced modifications did not. The quote actually transitions back to the Pending step, but none of the reasons are reset. It's likely that all of the Reasons in the Reset Reasons tab were unchecked, so the auto-Submit that the application performs after the Revise is actually auto-approving the quote and transitioning it back to the Approved step. Try to reset the reasons in the Revise action's Reset Reasons tab. After this, the action should function as expected on the user side.

How can Submit and its child actions by displayed on the transaction?

The Submit action and its corresponding Revise actions are the only actions that can be displayed on the transaction page to the end user. Approve/Reject and Request Approval actions cannot be displayed independently on the UI and are triggered incidentally.

- **Submit Action:** This action is displayed on the transaction page provided that the current user has been given access to this action in the appropriate Participant Profile.
- **Revise Action:** Similar to Submit action.
- **Request Approval:** This action is performed automatically when the user clicks on the Submit action and there are reasons to approve.
- **Approve:** This action is performed when the approver clicks on the Approve icon on the My Approval attribute. The ful-laccessuser can control the display of the My Approval attribute through Participant Profiles and must make sure that all approvers are given access to this attribute.
- **Reject**: This action is performed when the approver clicks on the Reject icon of the My Approval attribute. The fullaccessuser can control the display of the My Approval attribute through Participant Profiles and must make sure that all approvers are given access to this attribute.

Can a transaction have multiple Submit actions?

Yes. A transaction can have multiple submit actions and can simultaneously go through multiple approval processes related to each Submit action.

How are Standard Approval Notifications different from other notifications in Commerce?

Approval Sequences	Transition Rules
Notifications are sent for the following: • Pending Approval Notification: Sent to an approver when the approval becomes act-ive. • Retraction Notification: Sent to approvers when a reason is rest and approvers are no longer need to approve. • Rejection Notification: Sent to the submitter when a reason has been rejected. • Final Approval Notification: Sent to the submitted when the transaction approved. • Final Rejection Notification: Sent to the submitter when the transaction is rejected.	Sent only on step transition.
Most approval notifications are not customizable. See Email Templates to customize pending	Completely customized by the fullaccessuser.

Approval Sequences	Transition Rules
approval notifications.	
Customized modifications written in BML.	NO BML access. Only smart tags are available to allow dynamic values in email.
Special attributes available based on Email Templates or by using BML to customize a notifications.	Special attributes available in smart tags: • Document Link • Submitted by

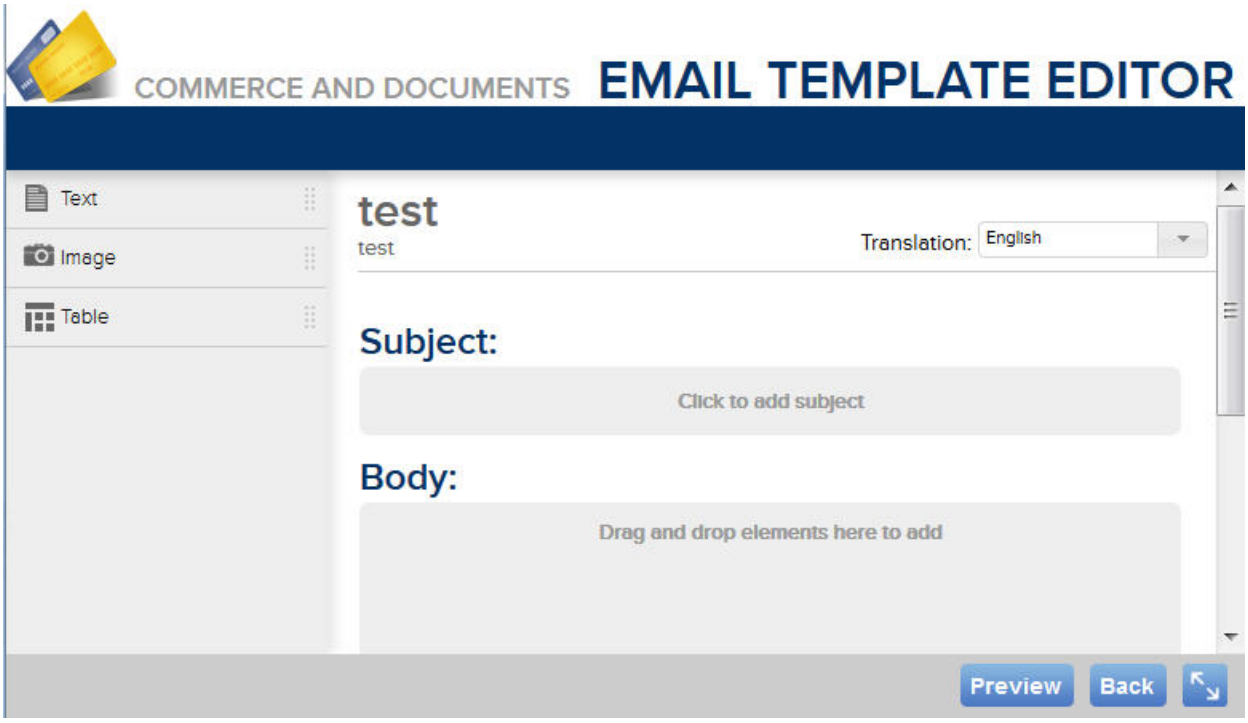
RELATED TOPICS

EMAIL TEMPLATES OVERVIEW

OVERVIEW

You can improve the look and feel of pending approval notifications with the use of email templates. You can create multiple, customized email templates that can be linked to or more approvers in an approval sequence. This feature also allows you to attach documents to the approval notification email.

Email templates can currently only be associated with pending approval notifications.

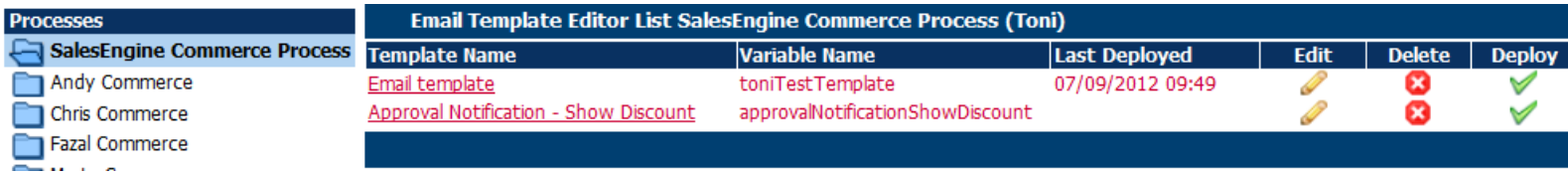


ADMINISTRATION

You can customize the look and feel of your pending approval notifications in an easy and intuitive way thanks to a slick visual editor and drag-and-drop behavior. They email template editor allows you to customize both the subject line and body of the notification(s). Some features allow you to add images, text and tables to the email body and quote-level commerce attributes to both the subject line and the body. Email Templates can be found under the Commerce and Documents section of the Admin Homepage.

Overview of Administration

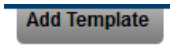
This overview begins on the Email Template List page for a given commerce process.



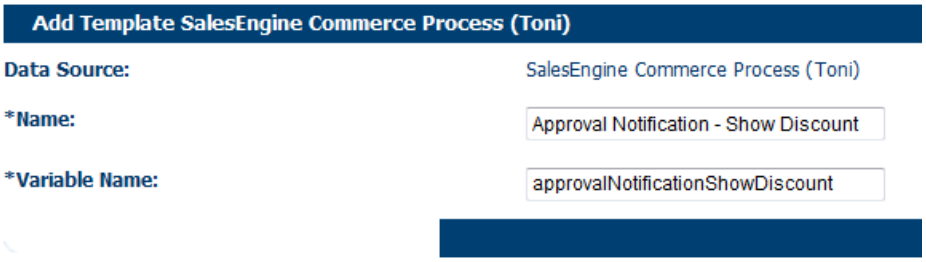
In the left-most panel, you'll see the header *Processes*. This is a list of the commerce processes available within your application. In the right-most panel, you'll see the header *Email Template Editor List*. The right panel is where you will perform the majority for your actions from adding, editing and deploying templates.

Icon	Description
	Click the pencil icon to edit the name and description of the email template.
	Click the delete icon to delete the email template.
	Click the deploy icon for a selected email template to deploy them one at a time. Un-deployed templates will be displayed in red. Note: Email templates are not available for use until they are deployed.
List	List Click the 'List' link under References to see which commerce process, action, and approval reason where the selected email template is being used.

Creating an Email Template

1. Navigate from the Admin Homepage > Commerce and Documents > Email Templates.
2. Select a **commerce process** from the left panel.
3. Click **Add Template**.
4. 

5. Enter a **name and variable name** for the email template.



6. Click **Add**. You will be returned to the Email Template Editor List page. See image in Overview of Administration section above. *You can click Back to return to the list page without saving.*



From here, you'll need to customize your email template. You should begin by editing the subject.

Translations

BigMachines offers multi-language support within a given template. BigMachines will create a copy of the original template in a new language, allowing you to manage content in different languages. Approvers will receive their pending approval notifications in the default language defined in their user profile, if an email template has been created in that language.



If you have a single language site, you will not have access to translations.

NOTES



Your operating system and email client will need to same fonts installed in order for them to render properly on the page.

A separator (~) will be displayed between the float attribute value and the unit (only when the float attributes have units).

Elements in the email templates will auto-save.

Adding, removing or reorganizing content is only available to the base language of the site.



Email templates are available for each commerce process.

Email templates are migrated when a commerce process is migrated. They cannot be migrated separately from the commerce process.

While BML is not required, advanced notifications are still available for use, if needed.

When a user loses internet connection, the application will automatically attempt to resend the notification and upon failure, will give the option to resend.

Email translations affect language only. Currency attributes will not be affected and will reflect the session's currency.



Image height and width settings do not always render properly in Microsoft Outlook.

Commerce HTML attributes and File Attachment attributes are not available for use in email templates.

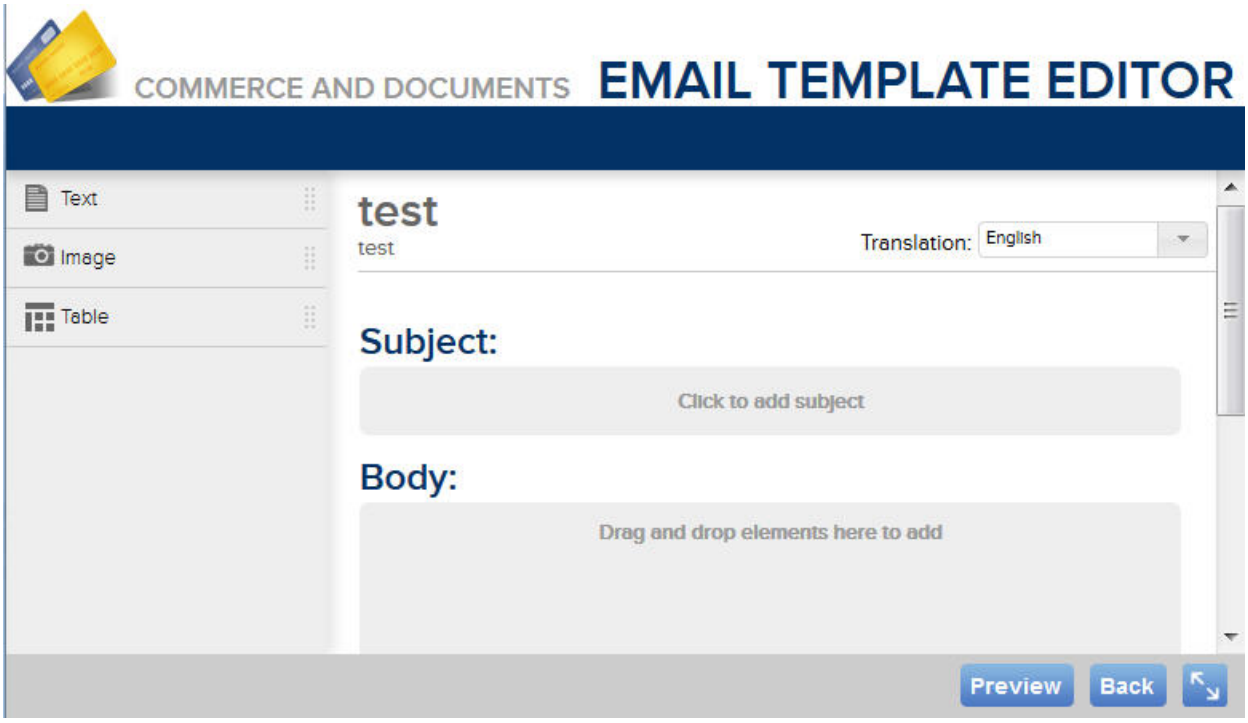
RELATED TOPICS

EMAIL TEMPLATES OVERVIEW

OVERVIEW

You can improve the look and feel of pending approval notifications with the use of email templates. You can create multiple, customized email templates that can be linked to or more approvers in an approval sequence. This feature also allows you to attach documents to the approval notification email.

Email templates can currently only be associated with pending approval notifications.

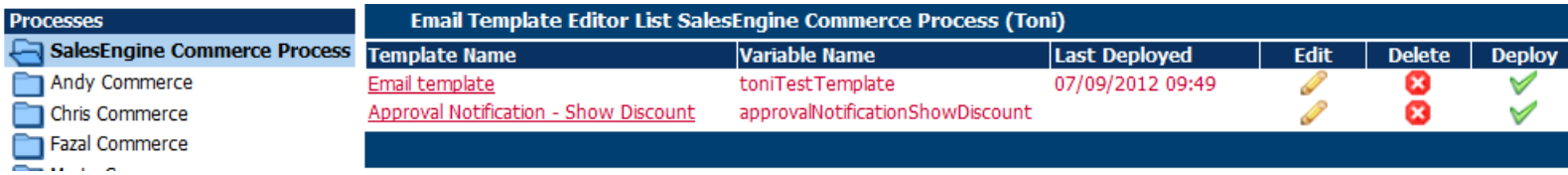


ADMINISTRATION

You can customize the look and feel of your pending approval notifications in an easy and intuitive way thanks to a slick visual editor and drag-and-drop behavior. They email template editor allows you to customize both the subject line and body of the notification(s). Some features allow you to add images, text and tables to the email body and quote-level commerce attributes to both the subject line and the body. Email Templates can be found under the Commerce and Documents section of the Admin Homepage.

Overview of Administration

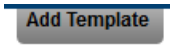
This overview begins on the Email Template List page for a given commerce process.



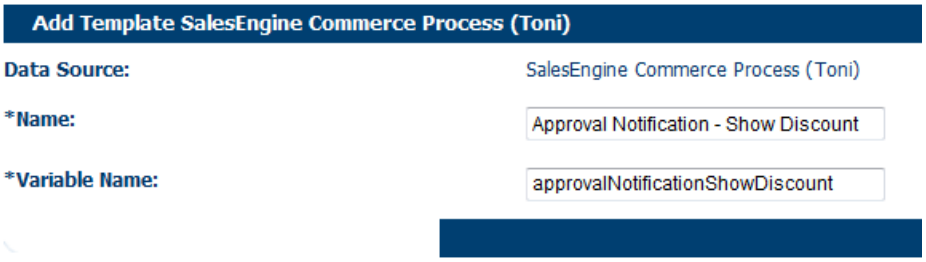
In the left-most panel, you'll see the header *Processes*. This is a list of the commerce processes available within your application. In the right-most panel, you'll see the header *Email Template Editor List*. The right panel is where you will perform the majority for your actions from adding, editing and deploying templates.

Icon	Description
	Click the pencil icon to edit the name and description of the email template.
	Click the delete icon to delete the email template.
	Click the deploy icon for a selected email template to deploy them one at a time. Un-deployed templates will be displayed in red. Note: Email templates are not available for use until they are deployed.
List	List Click the 'List' link under References to see which commerce process, action, and approval reason where the selected email template is being used.

Creating an Email Template

1. Navigate from the Admin Homepage > Commerce and Documents > Email Templates.
2. Select a **commerce process** from the left panel.
3. Click **Add Template**.
4. 

5. Enter a **name and variable name** for the email template.



6. Click **Add**. You will be returned to the Email Template Editor List page. See image in Overview of Administration section above. *You can click Back to return to the list page without saving.*



From here, you'll need to customize your email template. You should begin by editing the subject.

Translations

BigMachines offers multi-language support within a given template. BigMachines will create a copy of the original template in a new language, allowing you to manage content in different languages. Approvers will receive their pending approval notifications in the default language defined in their user profile, if an email template has been created in that language.



If you have a single language site, you will not have access to translations.

NOTES



Your operating system and email client will need to same fonts installed in order for them to render properly on the page.

A separator (~) will be displayed between the float attribute value and the unit (only when the float attributes have units).

Elements in the email templates will auto-save.

Adding, removing or reorganizing content is only available to the base language of the site.



Email templates are available for each commerce process.

Email templates are migrated when a commerce process is migrated. They cannot be migrated separately from the commerce process.

While BML is not required, advanced notifications are still available for use, if needed.

When a user loses internet connection, the application will automatically attempt to resend the notification and upon failure, will give the option to resend.

Email translations affect language only. Currency attributes will not be affected and will reflect the session's currency.



Image height and width settings do not always render properly in Microsoft Outlook.

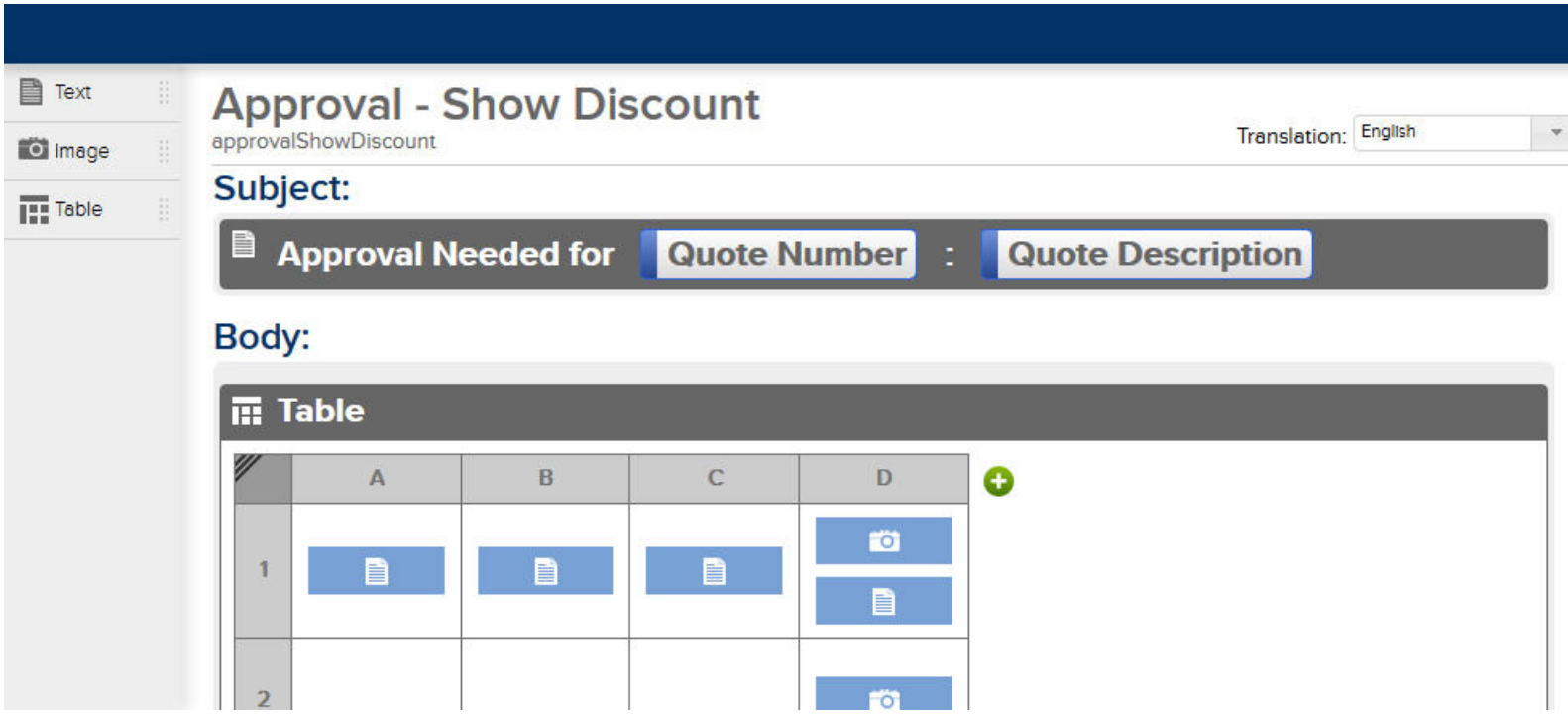
Commerce HTML attributes and File Attachment attributes are not available for use in email templates.

RELATED TOPICS

EMAIL TEMPLATE TABLES

OVERVIEW

BigMachines has enhanced the Email Template feature to allow fullaccessusers to construct tables within the templates. You can easily add tables to template body, as well as elements like text and images, to the table cells using drag-and-drop.



ADMINISTRATION

These instructions begin once an email template has been created. Please see Body of Email Templates for more details.

Managing Email Template Tables

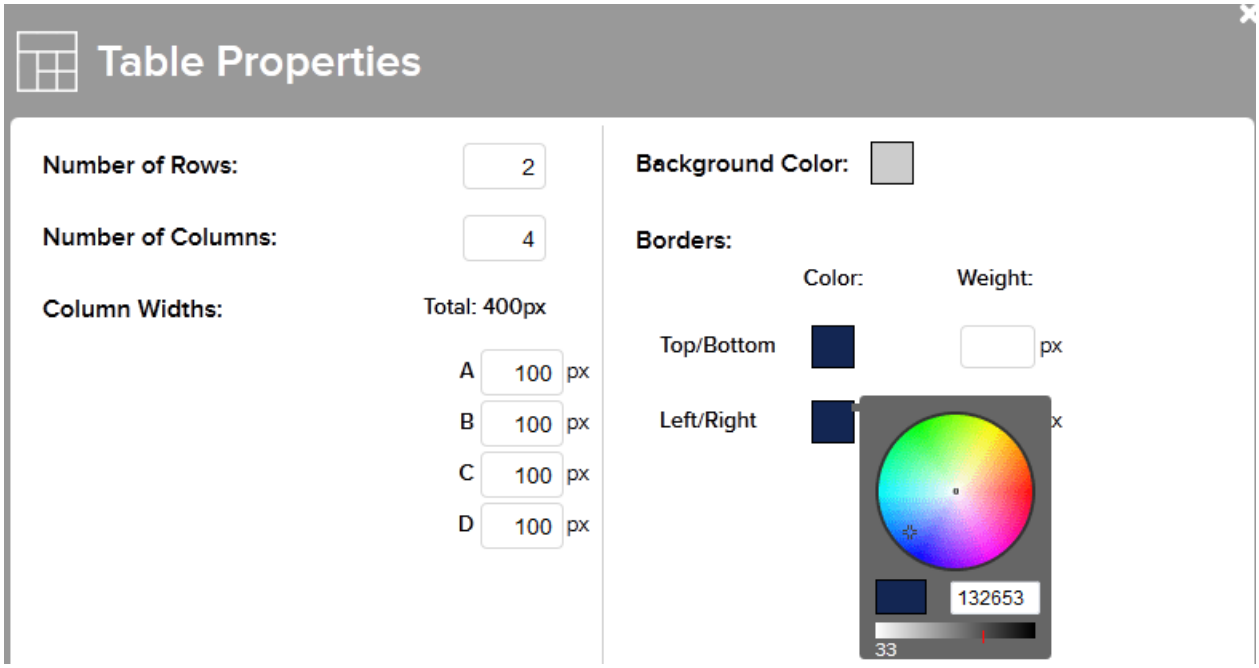
1. Click the **Table Component** and drag-and-drop one ore more tables into the email template body.

   The default table size is 1 row x 1 column.

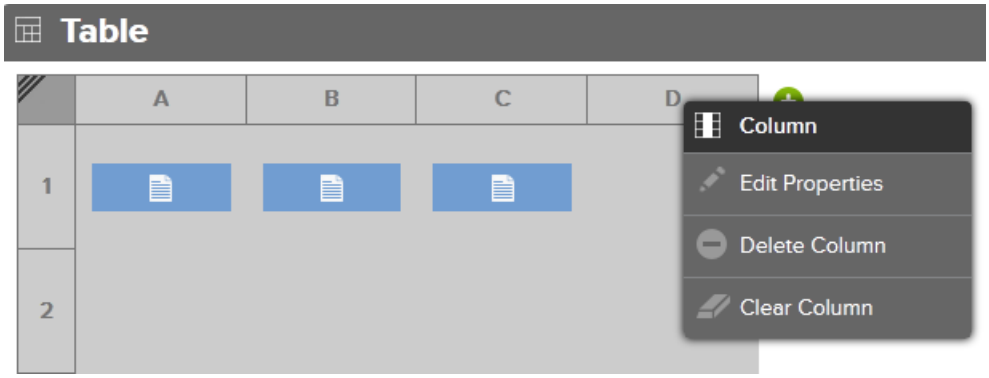
2. Click the icon on the **Table editor** to open Table Properties, where you can set the following:
   - **Number of Rows:** Enter an unlimited number of rows.
   - **Number of Columns:** Enter an unlimited number of columns.
   - **Column Widths:** Enter the width of each column in pixels.

- **Background Color:** Set the background color for the table. Please note that cell, row and column properties take precedence over this value.
- **Borders:** Set the color and width (in pixels) for the Top/Bottom and/or Left/Right borders. Please note that cell, row and column properties take precedence over this value

> Top/Bottom borders apply to all horizontal lines and the Left/Right borders apply to all vertical lines in the table.

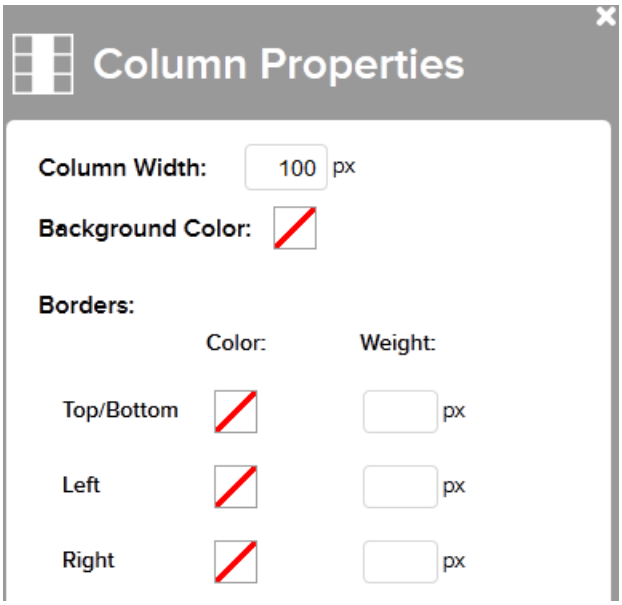


3. Click **Save or Close**.
4. Click the icon to add rows or columns directly on the table UI.
5. Click on a **row or column header** to open a tooltip menu.



6. Choose a tooltip menu item to:

   - **Edit Properties:** Edit the row/column size (in pixels), background colors, and border options.



   - **Delete Column:** Delete the row/column and all of it's contents.
   - **Clear Column:** Clear the contents from the row/column.

7. Click an **image or text element** to drag-and-drop it into table cells.



When hovering over a possible drop location, a placeholder will indicate the viable location, within that cell, where the element may be placed.

8. Click an **element** with a cell to open it's tooltip menu. Choose a tooltip menu item to:
    - **Text Menu:** Available only when a text element is selected.
        - **Edit Properties:** Opens the text editor where you can enter static text and/or attributes. Click here for more details.
        - **Delete Text:** Deletes the text element and it's contents from the table cell.
    - **Image Menu:** Available only when an image element is selected.
        - **Image Properties:** Opens the image editor, where you can edit the Source URL and image height/width (in pixels or percentages).
        - **Delete Image**: Deletes the image element from the table cell.
    - **Table Cell Menu:** Available in both text and image tooltip menus.
        - **Edit Properties:** Opens the Cell Properties pop-up where you can edit the cell background color and border color (s) and widths (in pixels).
        - **Clear Cell:** Deletes all elements, and their contents, from the table cell.

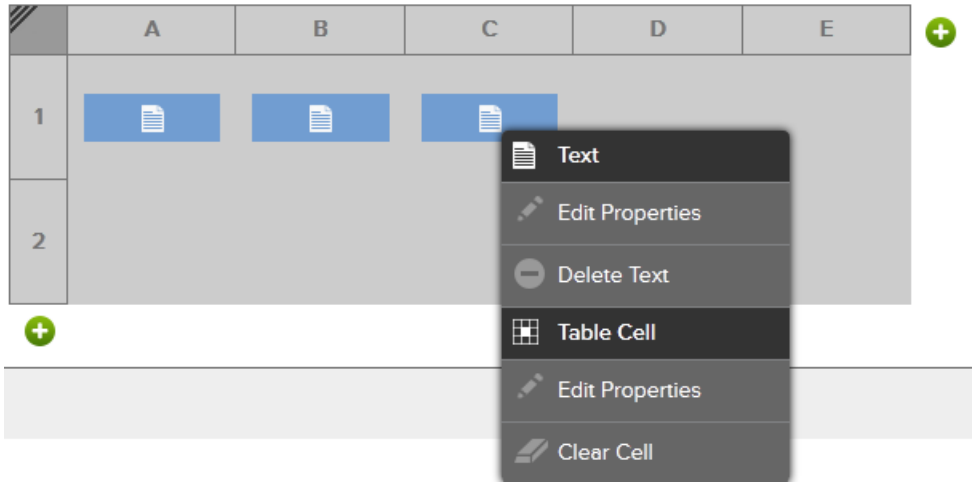
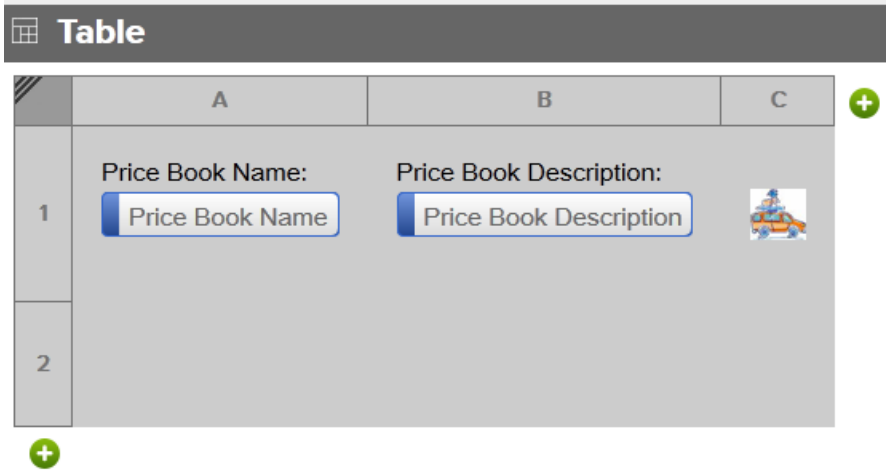


Table View With Content Included:



NOTES

Click the icon to delete the entire table and all of it's contents.

Drag-and-drop content from one cell to another.

Once content has been added to a text or image element within the table, the content will be visible in the table UI.

The precedence of properties is: Cell > Row > Column > Table.

Width (%) will display the image that percentage wide, relative to the column it is in. So, in a column w/ 100px, image width 50% would make the image 50px width.

Height (%) does not affect the image. It will stay at its native height because HTML tables have no height definition in rows, so there's nothing for the % to work against.

Adjacent borders on the same level will overwrite each other. For example, if you edit the border of cell A1, it will change the left border of B1, because they are the same.

Image height (%) functions differently than if the element was placed directly into the template body.

RELATED TOPICS

SUBJECT LINE OF EMAIL TEMPLATE

OVERVIEW

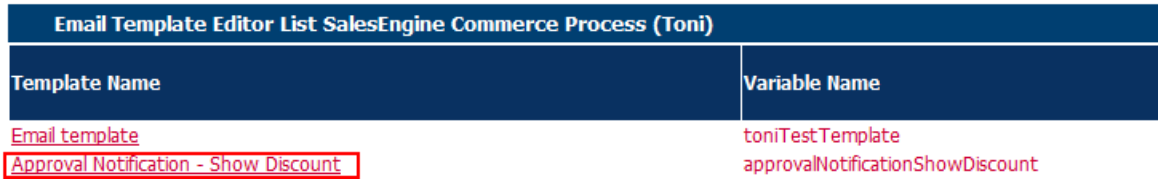
This article will show you how to edit the subject line of an email template.

ADMINISTRATION

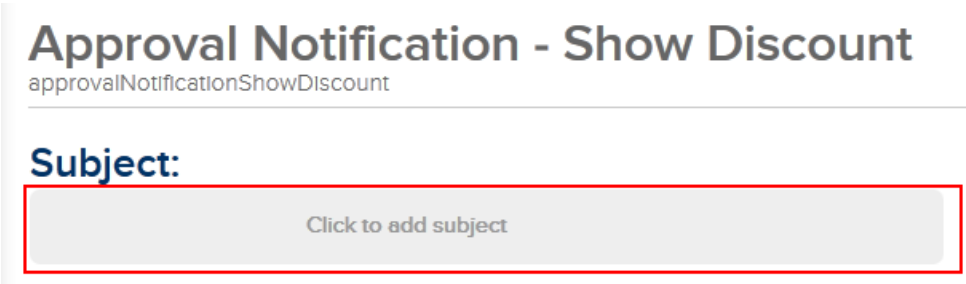
The steps begin from within the Email Template Editor List page.

Managing the Email Template Subject Line

1. Click on an **Email Template Name** to open the visual editor.



2. Click within the **gray Subject area** to begin populating the subject line. A pop-up window will appear for editing.



3. Select a supported language from the Translation drop-down menu.

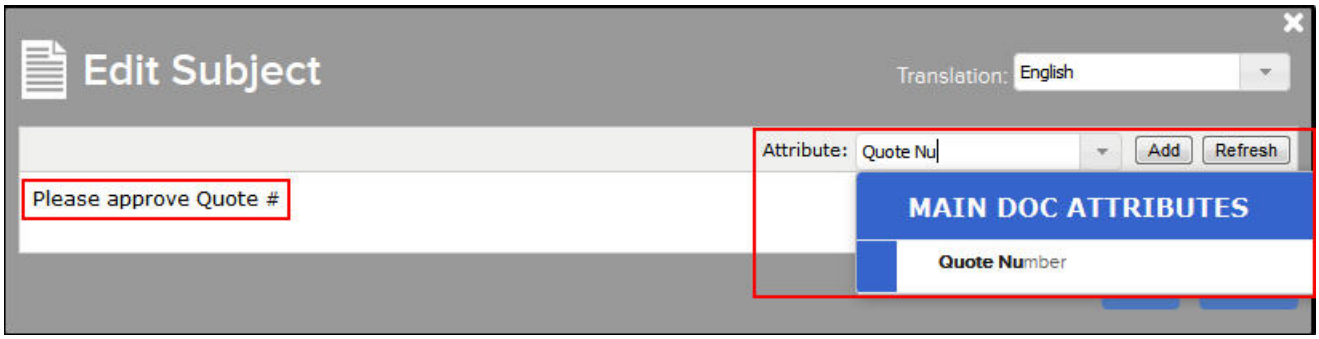


This feature is only available on multi-language support sites. Once you have defined a translation for a different language, the system will create a duplicate version of the template, in the new language.

Dynamic data can be added independently to each translate version. Approvers will receive the pending approval notification email based on the defined language preference on the user profile.

4. Type in **static text** to appear in the subject line.

5. Type in the filter OR use the Attribute drop-down menu to select one or more main-document attribute(s) and click add to include it in the subject line.



6. Click **OK** once you've finished customizing the Subject Line for the email template. You'll see the Subject area is now populated in the visual editor. Note: Click Cancel to return to the visual editor without saving your work.

Once you've created your subject, you can either Save the template or you can begin to edit the body of the email template.

NOTES

Click the Refresh button to clear attributes from the filter.

Only main-document attributes can be used in the subject.

There is a 'revert to base language' version. Any change that is made in the base language, as long as it has been reverted, will be available to the user that has the language.



Email translations affect language only. Currency attributes will not be affected and will reflect the session's currency.



Tables cannot be added to the subject line.

RELATED TOPICS

BODY OF EMAIL TEMPLATES

OVERVIEW

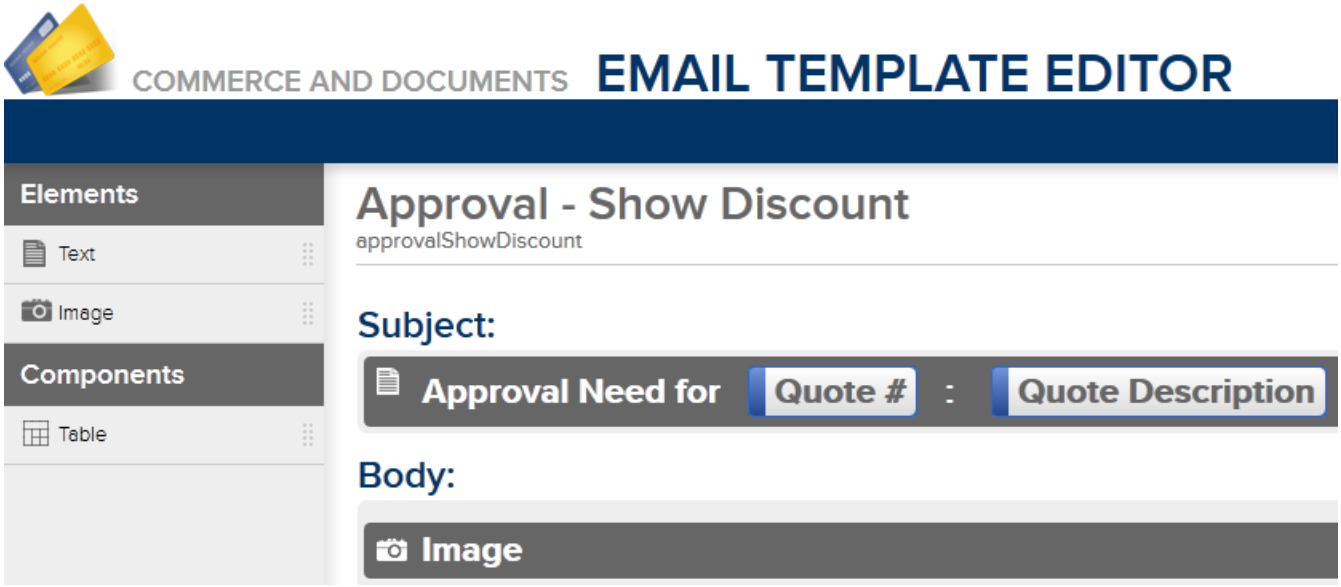
Create pending approval notifications from email templates. This article will show you how to edit the body of an email template.

ADMINISTRATION

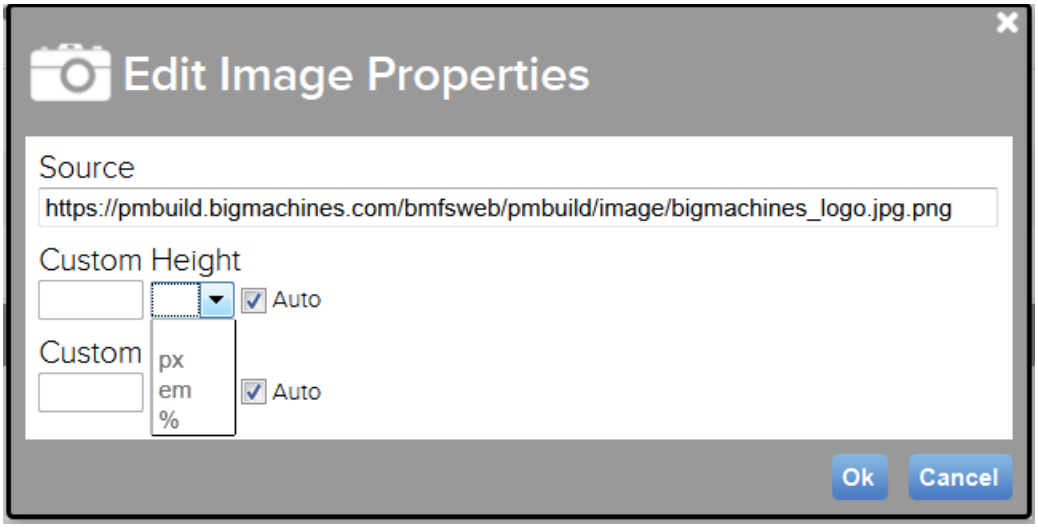
The steps below begin within the Email Template Editor List page.

Image Element

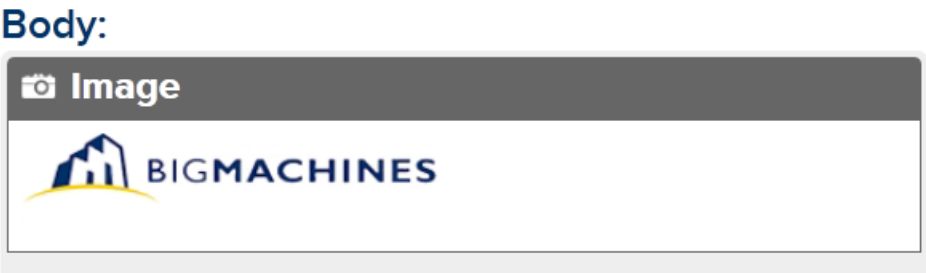
1. Click on an **Email Template Name** to open the visual editor.
2. **Drag and Drop** an image and/or text area into the **Body** of the email template. In this example, we'll start with an image. There will be a dotted line that appears when you drop the image into the body.



3. Click on the  icon to open the Edit Image Properties window. Do the following:
    - Enter the **URL of the image**, either by copying it from the File Manager or from an external URL, into the Source line.
    - Set the Custom Height and Width:
        - **Auto:** Select this checkbox to allow the application to set the image height and width.
        - **Custom:** Set the height and/or width in pixels (px), or percentage (%).



4. Click **OK** to save the image properties and return to the visual editor, or click Cancel to return to the visual editor without saving changes.



Text Element

1. Drag and drop a Text section into the Body area (see Step 2 in Image Element section) and click the icon to open the Edit Text window.

2. Select a supported language from the Translation drop-down menu.

> This feature is only available on multi-language support sites. Once you have defined a translation for a different language, the system will create a duplicate version of the template, in the new language. Dynamic data can be added independently to each translate version. Approvers will receive the pending approval notification email based on the defined language preference on the user profile.

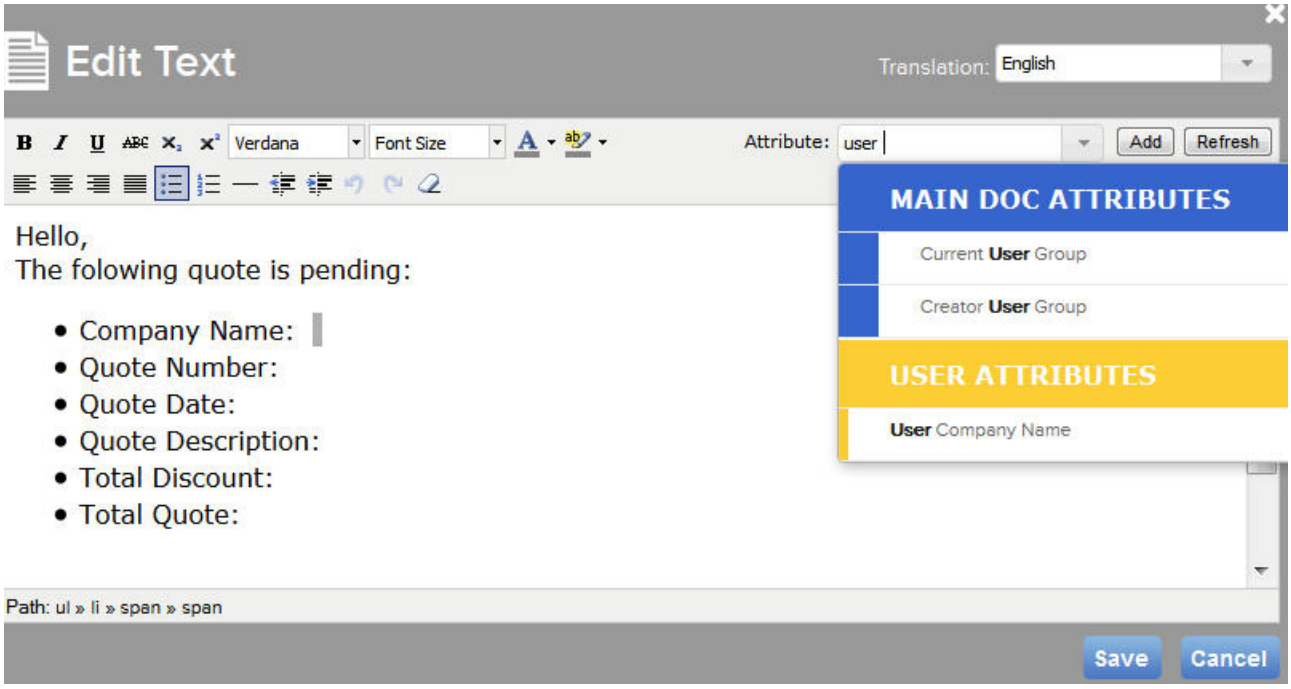
3. Use the text editor to add content to the body of the email template. Note: The Edit Text window is a rich text editor that will also allow you to add User and Main-Document(Quote Level) Attributes to the body of the email.



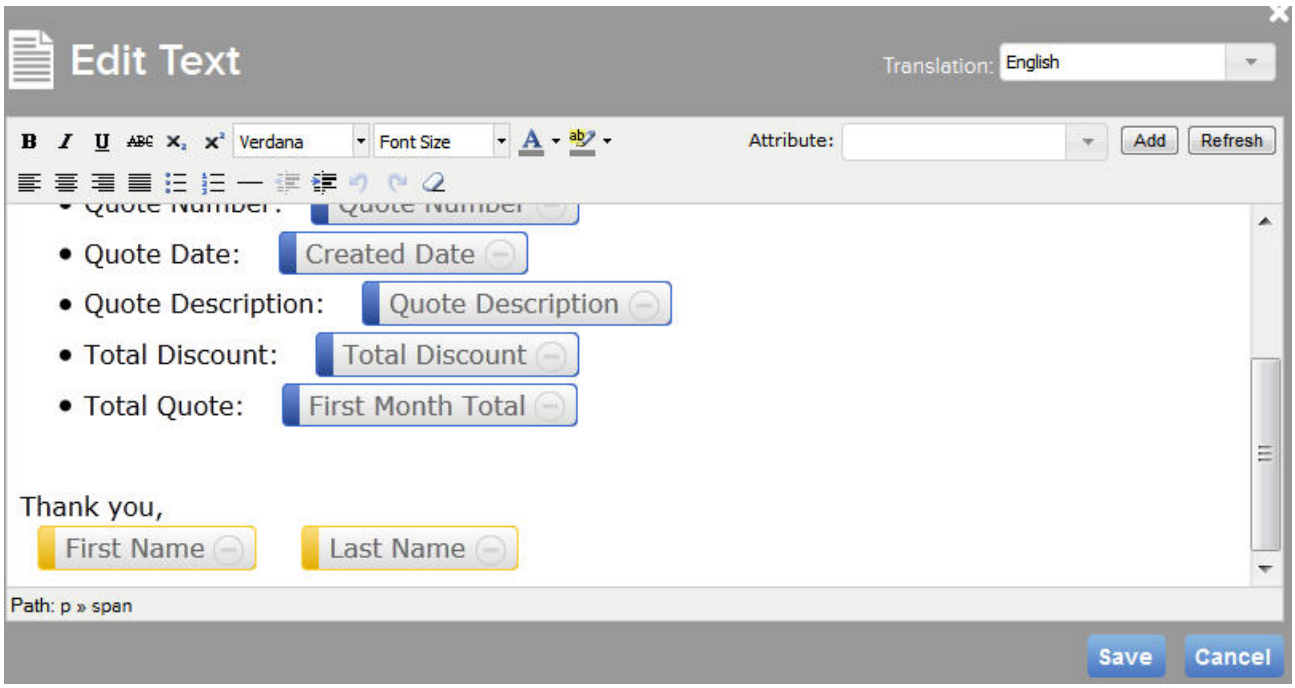
Use the following options to format the rich text content in the editing area:

Element	Icon	Description
Font Formatting Icons	B I U	Select text and click bold icon (B), italic icon (I), or underline icon (U).
Strike-through	ABC	Select text and click strike-through icon to place a line through the selected text.
Sub and Super-script	x_2 x^2	Select the text to be modified and click the superscript or subscript button to apply formatting.
Font Family	Verdana	Select a font-family from the drop-down menu. The menu contains all supported fonts. Note: The users computer and/or email client will need the same fonts in order for the email to render properly.
Font Size	Font Size	Select a font size using the drop-down menu. All font sizes are defined in points (pt).
Font Color	A	Select text and use the drop-down menu to select a font color.
Highlight	ab	Choose a color to highlight selected text.
Attributes	Attribute: Add Refresh	Use filter and/or drop-down menu to add user or main-document attributes to the email body.
Alignment		Select text and choose Align Left, Align Center, Align Right or Justify to apply to the text.
Bullets or Numbered List		Select text and cick the bulleted list or ordered list icon to apply formatting.
Horizontal Line	—	Place your cursor in the editor and click horizontal line icon to add it to the template.
Indent and Out-dent		Select the text to indent or outd-ent, or place your cursor on the line that needs to be indented/out-dented.
Undo and Redo		Click the undo button to undo the last action. Click the redo button to redo an action you've undone.
Eraser		Select text and click the eraser icon to remove text formatting.

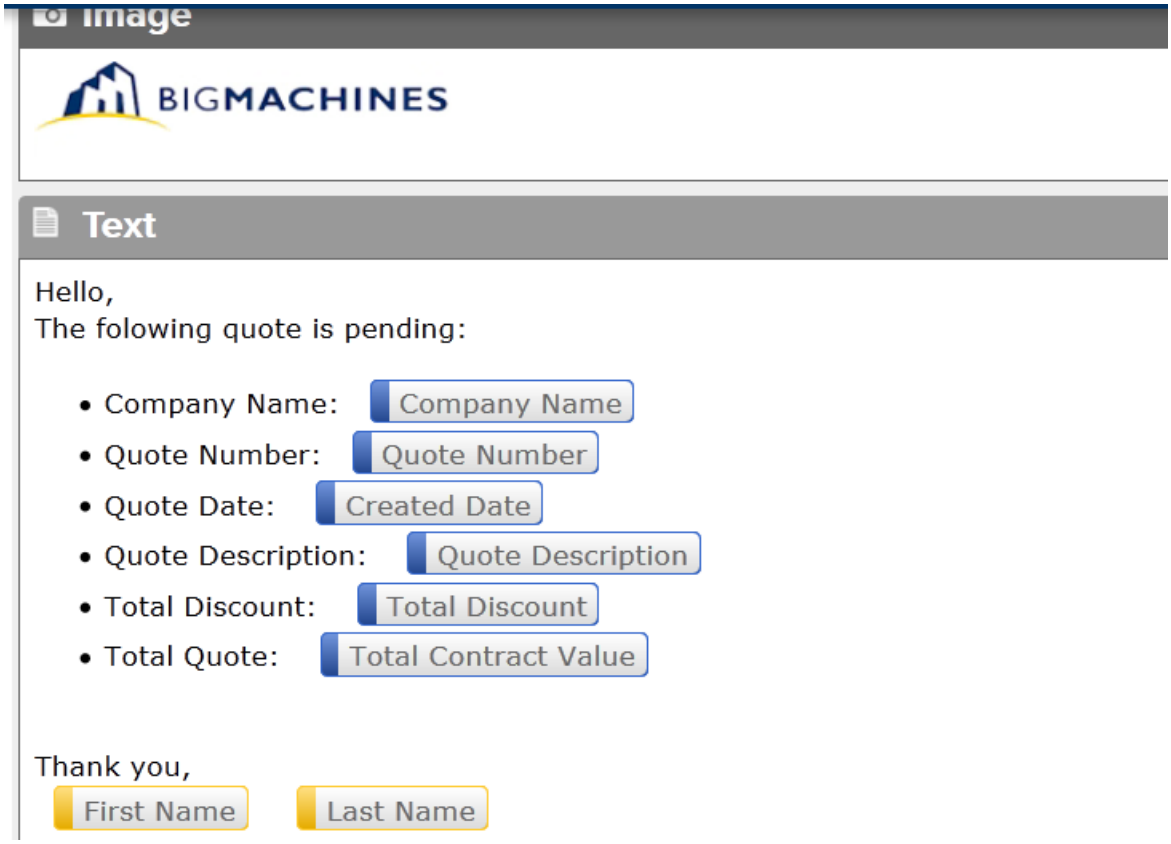
Example 1:
In this image, static text has been added to the email body, and the user is selecting an attribute to add to the body.



4. Click **Ok** after adding all of the required text and attributes to the email body. Note: Click Cancel to return to the visual editor home page without saving changes.



5. Now that you've completed the body of the email, you'll see the following on the visual editor home page:



6. Click one of the following:
    - Click **Preview** to see a preview of your email template. Note: You will be prompted to enter a Transaction ID.
    - Click **Back** to return to the Email Template Editor page without saving changes.
    - Select **the arrow icon** to toggle to a full-screen view.
7. Click the **deploy icon** from the Email Template Editor List page in order to make your template available for use for pending approvals.

NOTES

Please refer to the article Email Template Tables for more information on how to use Tables within the email template body.

There is a 'revert to base language' version. Any change that is made in the base language, as long as it has been reverted, will be available to the user that has the language.



Email translations affect language only. Currency attributes will not be affected and will reflect the session's currency.



HTML attributes are currently not supported.

You cannot add conditional blocks to the body of the email.

RELATED TOPICS

DOCUMENT ENGINE OVERVIEW

OVERVIEW

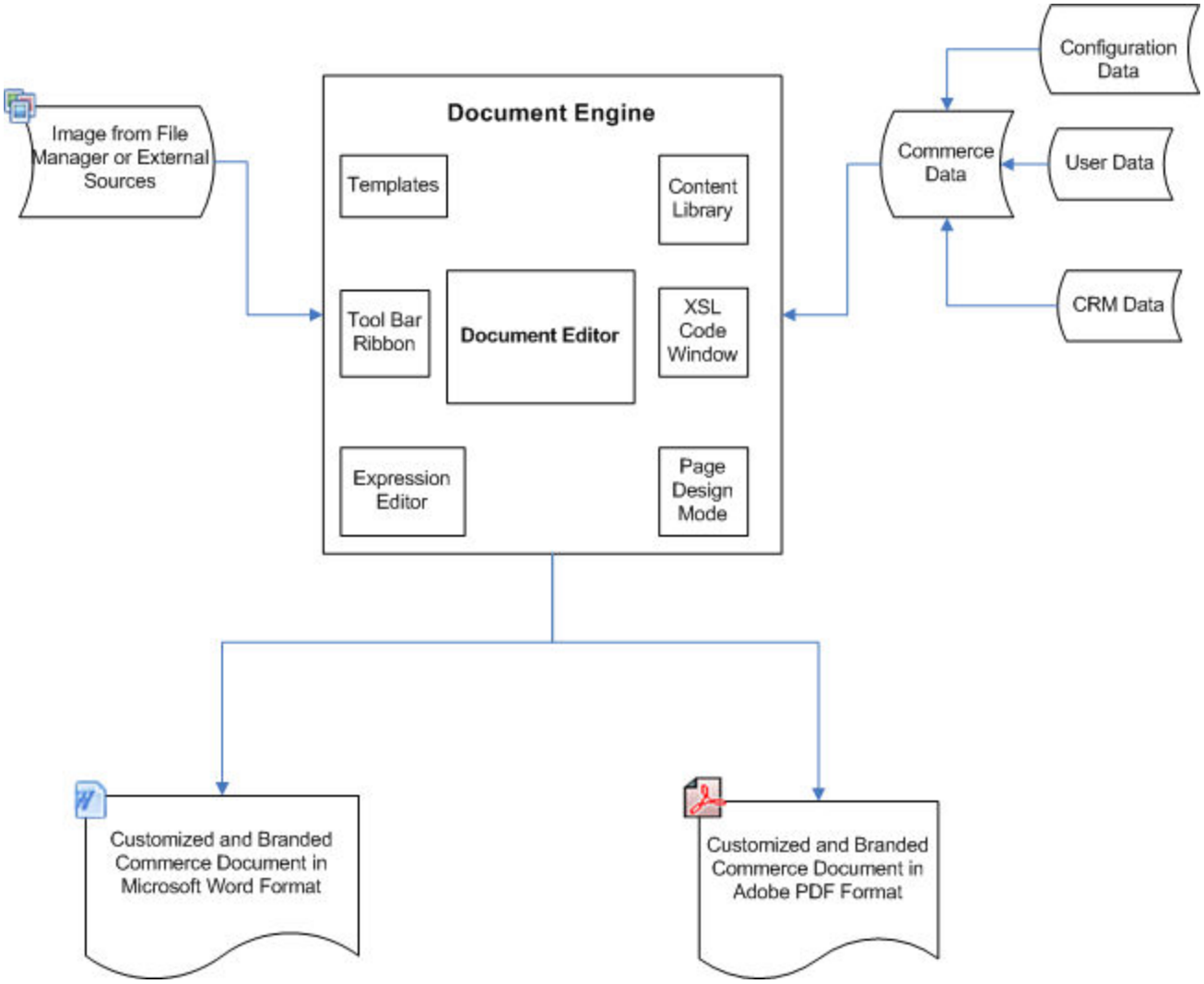
The Document Engine powers the BigMachines Proposal Generator and the BigMachines Contract Generator. Users can create professional and customer-friendly proposals and submittals as well as customize and brand proposals for specific products, channel partners or customers. It also automates the generation of customized contracts driven from approved and submitted quotes and orders.

Document customization capabilities:

- Populate proposals with relevant customer data
- Include detailed quotes as sub-documents or appended-documents
- Add customized terms and conditions based on configuration options
- Add customized and branded cover letters from user-defined and/or company-defined templates
- Add technical data sheets, pictures, and drawings
- Add quality documentation or industry specifications
- Browse and append any external documents from the user's desktop, company intranet, or internal shared drives
- Generate proposals, quotes or contracts in MS Word or Adobe PDF

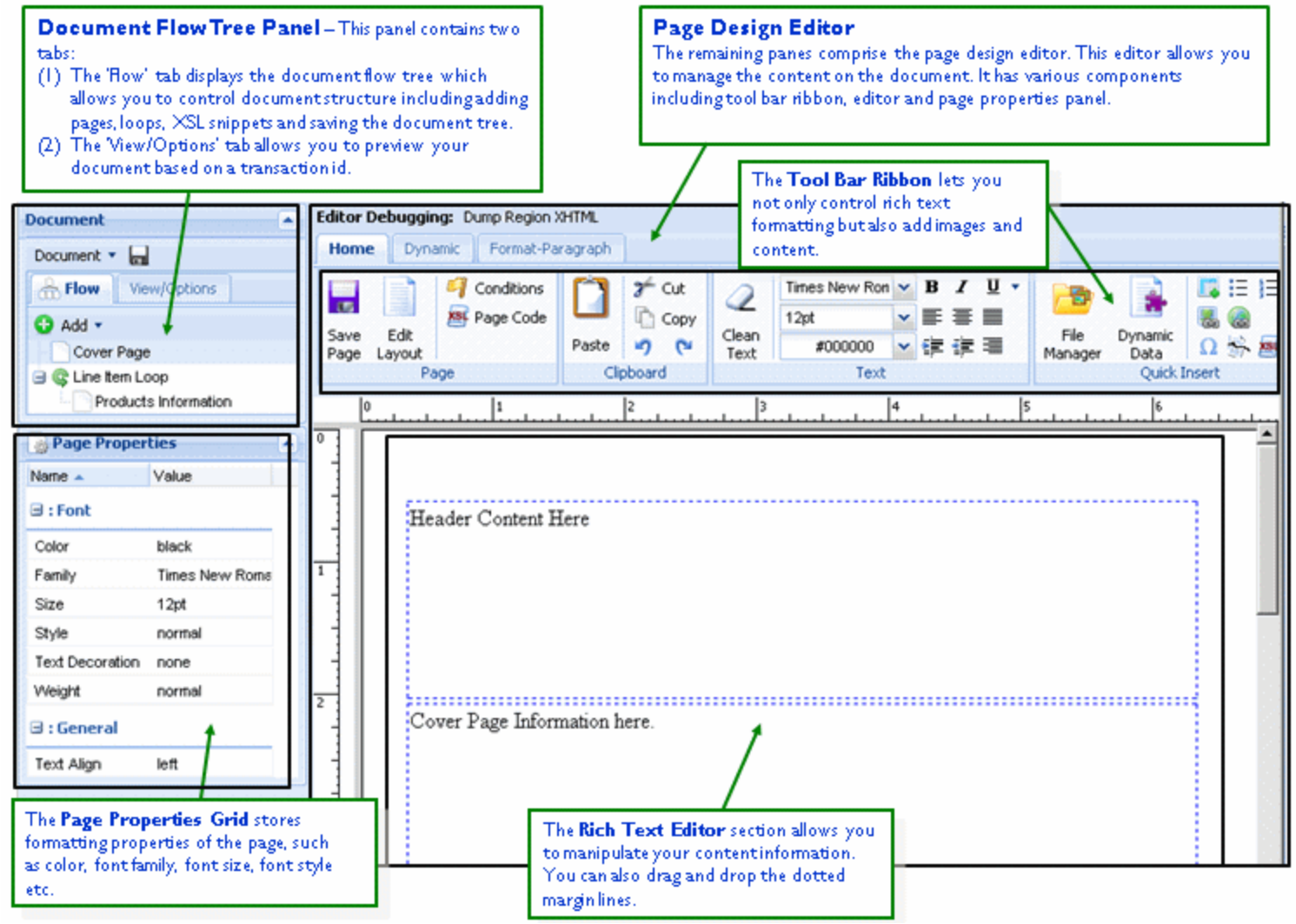
ADMINISTRATION

Document Engine Structure & Relationships



Document Editor

The BigMachines Document Editor is a visual XSL editor designed for authoring and editing XSL templates. Users can produce dynamic and customized commerce documents in either PDF or RTF formats. This visual XSL Editor helps combine the ease of a WYSIWYG tool with the power of XSL, facilitating quick and error-free template development. The document editor contains various components and features that make it so powerful and easy to use. Using a combination of the components and the features, you should be able to put together a custom proposal very quickly.



Key Components of Document Editor

Pages

In the Document Engine, a Page does not represent a literal page as your body content may end up flowing over to a new page when it's printed, depending on the data on the user-side. Think of a Document Engine Page as a place where you can define a constant header and/or footer with fixed dimensions. If your body content overflows to more pages, those pages will share the header/footer that you defined on the Document Engine Page. Additionally, you can insert hard page breaks into your page body content. In technical terms, a Page will output a complete `xsl<fo:page -sequence>` node.

Loops

Loops in the Document Flow Tree repeat over all line items and can display individual Product Information pages for each line item in the loop. You can edit the loop to control how the pages within a loop are generated. In technical terms, a loop in an `<xsl:-for-each>` which will ultimately be inserted outside of the <fo:page-sequence> tags that are generated for each Document Flow Page. Thus Document Flow Loops are useful for creating repeating pages.

XSL Snippets

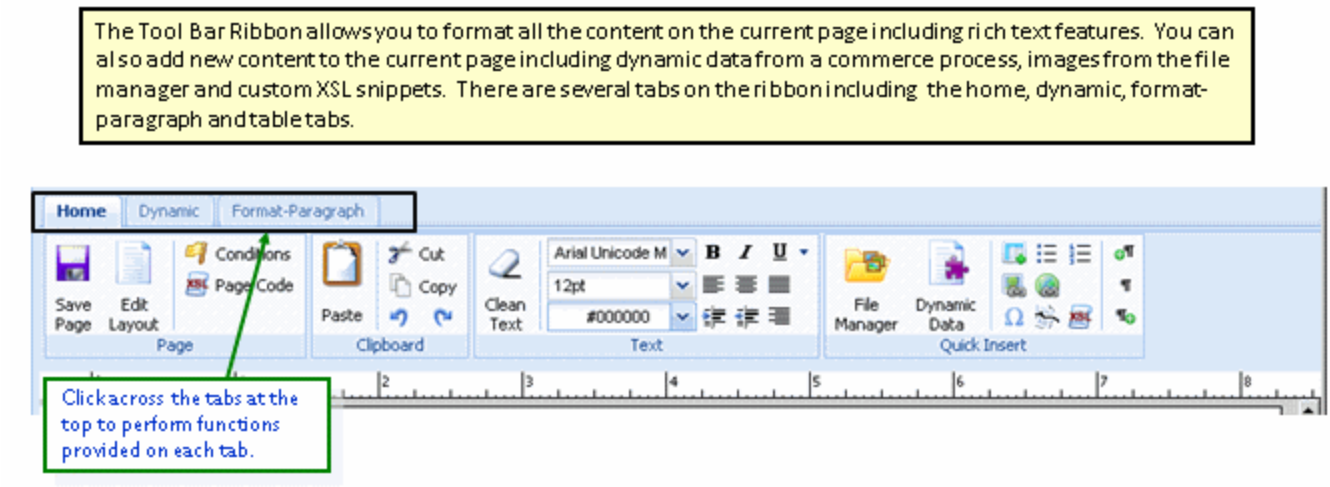
The code that you write in a Document Flow XSL Snippet will be inserted between the `<fo:page-sequence>` nodes that are generated by Page objects. Therefore, to avoid XSL errors, you should only write XSL that will either be `<fo:page-sequence>` nodes or generate `<fo:page-sequence>` nodes.

Features in a Document

- Enhanced Grid View section to view and edit the page properties
- Easy-to-use and rich user interface for Page Editing
    - Tabbed-ribbon tool bar
    - Page regions and margins can be set through dragging
- Access to the file manager directly from the Editor
    - Insert images and links from the file manager through drag-and-drop
    - Add and manager files using the User Interface
- New formatting features such as
- Nested lists, custom bullets
- Paste from Microsoft Word retains formatting
- Table Formatting
- Preview and debug the document to test, perfect and optimize the template before converting it
- At any time during the XSL template editing process, the user can transform the file(s) with a single click to preview the generated output as either PDF, Advanced PDF or RTF
- Edit the Global XSL snippet and maintain global variables, templates, etc.
- Customize the existing editor XSL data (such as <fo:simple-page-master>) in an advanced XSL code panel
- Quick and Easy Book Marking and Table of Contents

Tool Bar Ribbon

The Tool Bar contains actions that the user can perform on the elements of a Document Page in the Page Design Mode. The actions on the Ribbon are grouped in tabs. CliSelectingcking on a tab switches the tool bar actions to a different set of actions. For more information, see Toolbar.



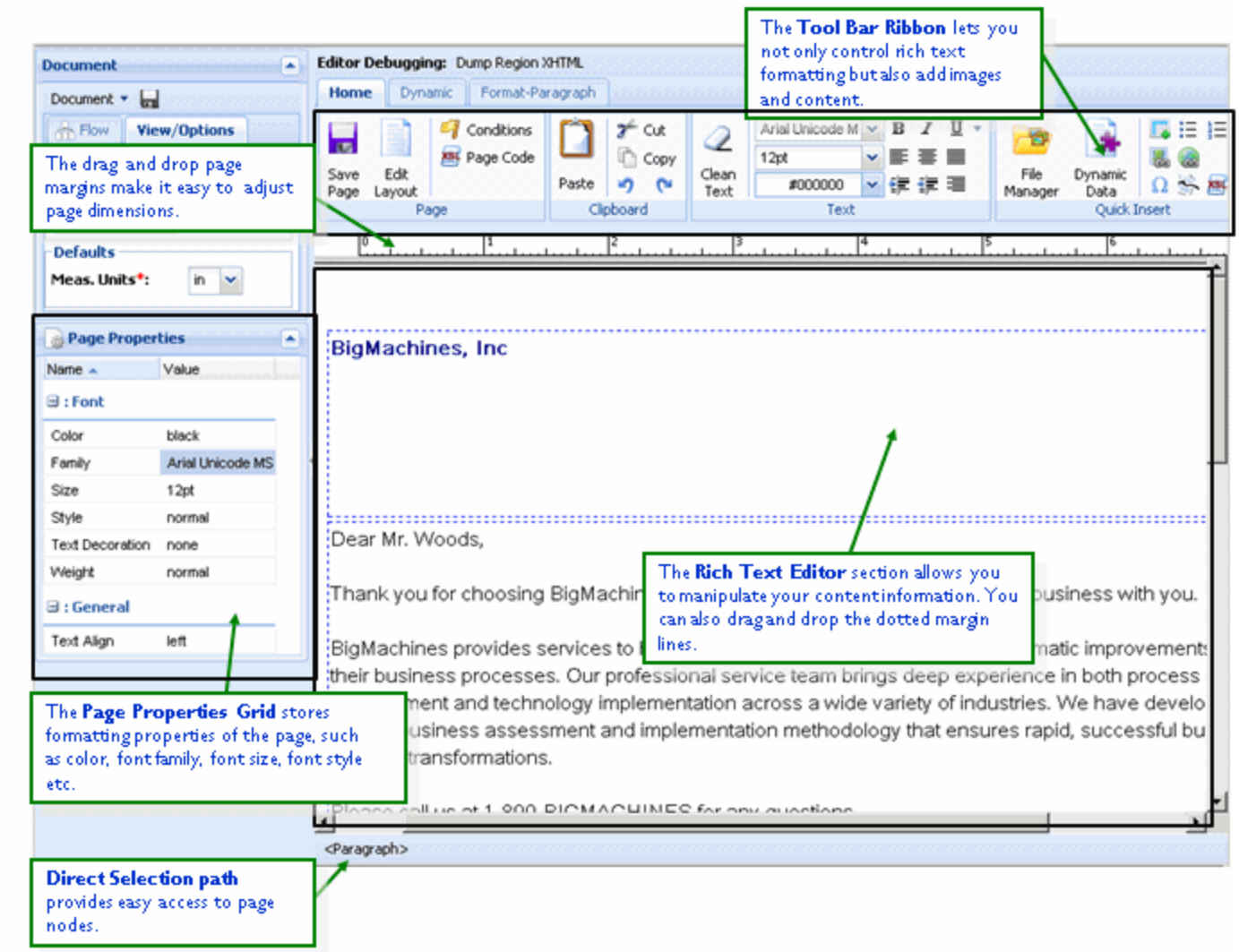
The visibility of tabs on the ribbon depends on the current active page region. **Example:** Table tab is only visible when the cursor is inside a table. In addition, tool bar items within a tab can be enabled or disabled based on whether it is appropriate for the current selection. **Example:** borders are only applicable to block nodes like paragraphs, tables, etc. and not for in line elements. In this case, the border actions will be disabled if the user is currently editing an in line selection.

Available Tabs on the Tool Bar Ribbon

- **Home** - This is the default tab, which contains the most common actions. The Home tab is always available in the Page Design Mode. See Home Tab Actions.
- **Block Format** - The Block Format tab is always available in the Page Design Mode. The actions on this tab allow the user to format the selected node.
- **Dynamic** - The Dynamic tab includes actions that allow the user to customize their pages.
- **Table** - The Table tab allows the user to customize tables in a page. This tab is visible only when the selected node is a table.
- **Image** - The Image tab is visible only when the selected node is an image. The actions in this tab can be used to change the source URL and the height and width of the image. Since images have fixed widths, the user must specify the image size.
- **Hyper Link** - The Hyper Link tab is visible only when the selected node is a hyper link. You can use this tab to edit the hyper link.
- **Book Mark** -The Book Mark tab allows the user to change the name of the bookmark. This tab is visible only when the selected node is a bookmark. A bookmark name must be unique to a page.
- **List** - The List tab allows the user to customize lists in a page. This tab is visible only when the selected node is a list or a list element. You can create both ordered and unordered lists. BigMachines also allows you to create nested lists.

Page Design Mode

The Page Design Mode is where the user builds and edits Pages. This mode includes a collection of components for authoring and formatting text. The user can also insert dynamic data in the text, which is data that easily and automatically changes to allow for easy customization. Navigation: Document Application Panel > Document Flow Tree > desired page



Components of Page Design Mode

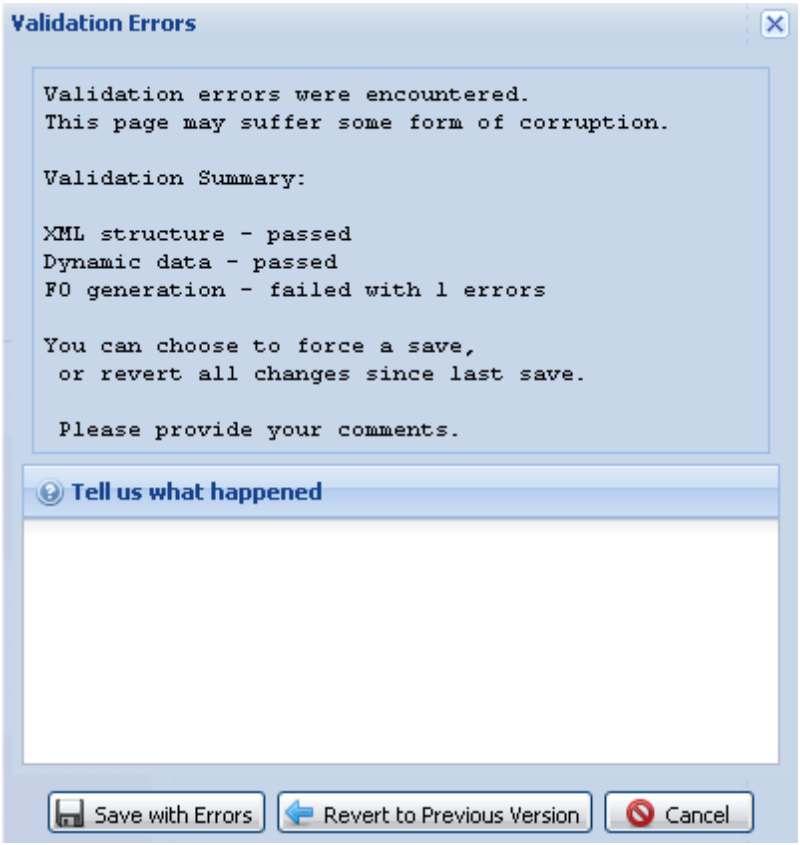
- **Page Properties Grid:** The grid displayed on the left panel stores formatting properties of the page, such as color, font family, font size, font style etc.
- **Page Design Tool Bar Ribbon:** The Tool Bar Ribbon comprises a menu of actions that can be used to edit and format the document pages. The tool bar consists of a set of tabs with groups of actions. Selecting a specific tab switches the tool bar to display the chosen tab with its menu items.
- **Rich Text Editor Region:** The user can view and edit the text within a page. The user can also choose to enter dynamic data using the Dynamic Content Window. The action on the Ribbon can be used to manipulate the content in this region.

- **Page Vertical & Horizontal Rulers:** The vertical and horizontal rulers provide a sense of the document's dimension for easier formatting. The rulers are defaulted to the Document Default Units option that can be changed in the Settings Tab of the Document Application Panel.
- **Draggable Page Layout/Margins:** The dimensions of the pages can be changed easily by dragging on the margins.
- **Direct Selection Path:** The user can select entire nodes in the RTE. This is useful in that the user can select the exact node to editing or formatting. Selecting a node button selects the entire node in the Rich Text Editor.

Validations & Optional Feedback

You can optionally run a validation on page data when saving in the Document Engine. When an error is found during validation, you will get an error message. This will allow you to revert to the last saved copy, cancel the save or save the corruption. You can send feedback to the BigMachines development team directly when you get the corruption. *Make sure to define the steps to reproduce.*

1. Navigate from: Admin Homepage > Commerce and Documents > Commerce Settings.
2. Select Yes for the option **Document Engine – Validate Page Data on Save**. *This is set to NO by default.*



Upon validation error, you can

- **Save with Errors:** Save despite any errors. From this point on, you will encounter this error on this content until it is fixed.
- **Revert to Previous Version:** Revert to the last saved state. This will not rollback changes made to content of dynamic items like XSL snippets, simple conditions, loops, clause conditions, row conditions/loops, formatted expressions and others.
- **Cancel:** Cancel the operation which caused this validation to fire. You will not lose changes you made.



The validation is run on complex conditionals.

The error message is sent to development, even if you don't provide feedback details. This feedback is NOT replacing immediate bug reporting, so turn around time for this would be much longer. Feedback is stored in a database that is periodically analyzed by development.



Revert to last save is not available for new complex conditionals. *This will be available once the complex conditional has been saved for the first time.*

Supported Node Types

Content in the Rich Text Editor is organized in a way similar to the way HTML is structured. This not only helps in providing structure to pages in a document, but also allows for enhanced functionality. **Example:** depending on the node type that is selected, specific ribbon tabs can be shown/hidden. This lets the user know exactly what actions can be performed on a given node type. The Structural Building Blocks of a document are known as Nodes.

Supported Node Type	Description
P	paragraph
Table	table
TR	row in a table
TD	cell in a table row
SPAN	inline style
IMG	image
A	anchor link
UL	list
Li	list item
LOOP	repeating block

Supported Node Type	Description
CONDBLOCK	condition block (if...then)
IFELSE	if else
DATA	dynamic data
BMK	bookmark
PGNUM	page number



Some Node Types share names with corresponding HTML tags with similar functionality. However, these are custom nodes are not to be confused with HTML tags. Node are strings, and each can be translated.

Best Practices

- While working in the Doc Engine Editor, click the  button to save your work.
- Use standard functionality over XSL when possible.
- Make backups.
- After modifying a document page through FireBug and saving the changes, reload/reopen the page before making any further changes. *This ensures that the html changes made using FireBug are properly reflected in the document XSL.*

TROUBLESHOOTING

Fixing corrupt documents

1. Corrupted file error messages are sometimes caused by deleting/altering tables, rows, or conditionals. *Follow these steps to rectify the situation:*
2. Open the corrupted file in Doc Engine.
3. Navigate to the area where an item was recently deleted/edited prior to the file corruption.
4. Open Firebug within Doc Engine.
5. Click the “Click an element in the page to inspect" button in the upper left hand corner.
6. Click the row (or table, paragraph w/ conditional) that was recently altered and may have caused the corruption. *The element you selected will be highlighted in the Firebug editor.*
7. Examine the element's HTML to determine if it is causing the problem.

8. Once the corrupted element has been located, right-click the element and select “Delete Element.” *You will need to rebuild the table (or row) that had been there.*



When deleting items in Doc Engine, begin with the innermost element (e.g. dynamic data, XSL snippets, complex conditionals) and then work outwards (e.g. row, table).



To avoid this problem in the future, delete elements properly by using the buttons provided within Doc Engine.

USE CASES

Use Case: Displaying a Conditional Watermark

Let's say you need to display a background "DRAFT" image for quotes that are in a pending state. We recommended that you define a global `main_doc` variable to point to your header attributes. You can only define it once in the document, so if `main_doc` is already defined, don't define it again.

1. Upload an image to the BigMachines File Manager to use as the watermark. *You will need to do this first in the BigMachines applications since you can't upload files through the Document Engine. Make sure that the image that you use is a valid image that the FO processor can read.*

2. Add the following to define a custom watermark image:

    - Replace the image URL to point to your watermark image.

    - Change the logic of the text to one that you need for your application. In the example below, we are checking for the main document commerce attribute with variable name `quoteStatus_q` and seeing if it equals the text "Pending". *Your implementation may be different. You can add more <xsl:when> statements if you need to.*

    - An `<xsl:choose>` statement is basically an `IF/ELSE IF/ELSE` block. *The processor will retrieve the value of the first <xsl:when> conditional statement that is true. In this case, we are returning the URL of the watermark if the watermark should be displayed.*

```
<xsl:variable name="main_doc" select="/transaction/data_xml/document[@data_type=0]"/>

<xsl:variable name="watermark">

<xsl:choose>

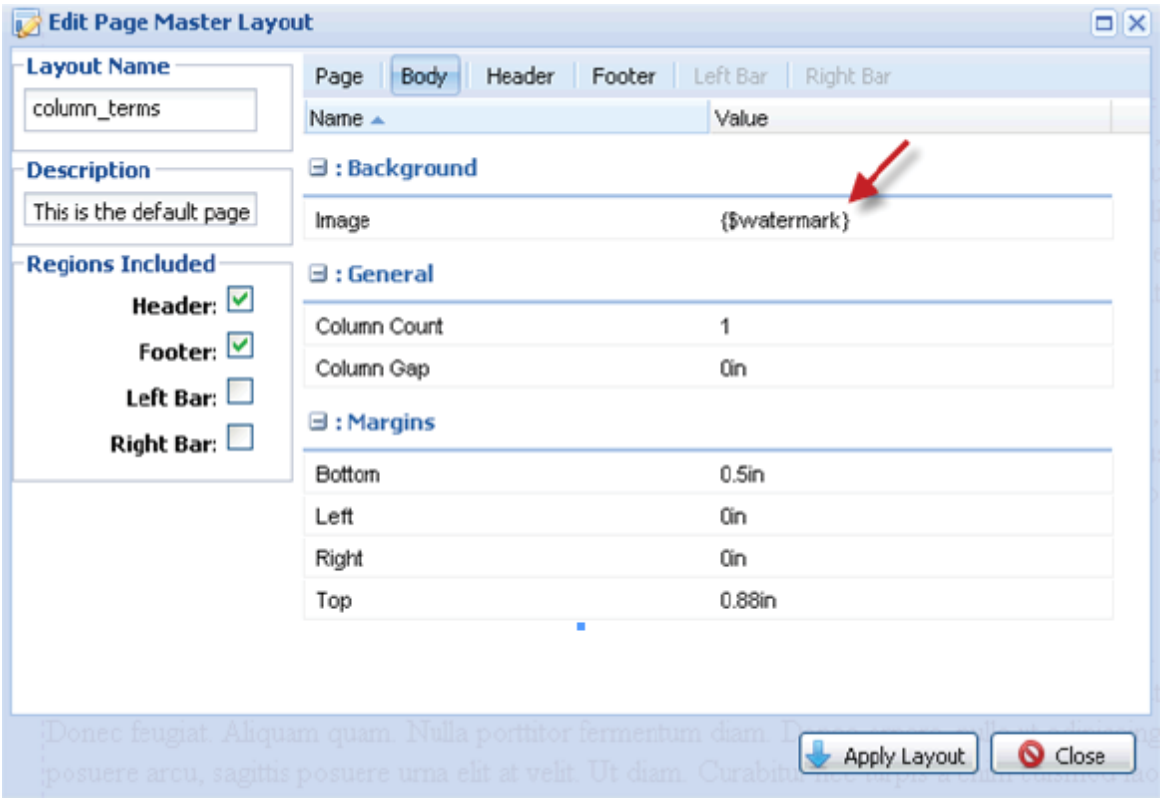
<xsl:when test="$main_doc/quoteStatus_q = 'Pend-
ing">/bmfsweb/pm/image/Proposal/watermark.jpg</xsl:when>

<xsl:otherwise></xsl:otherwise>

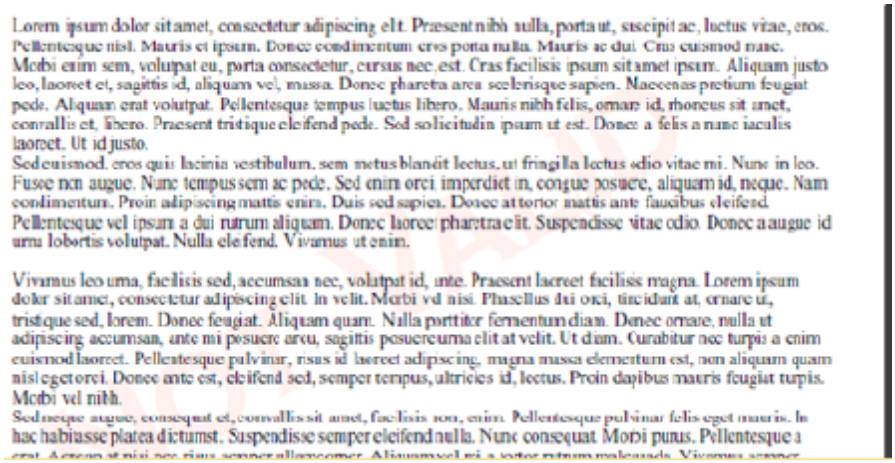
</xsl:choose>
```

```
</xsl:variable>
```

3. Save your **Global Snippet** AND save the document.
4. Open the page where you want to place the watermark. Select **Edit Layout** from the Home Tool bar to open the Layout Editor.
5. Click on the **Body** tab and go to the image property. *You will tell this page layout to read the global watermark variable.* Enter `{$watermark}` into the Image field. It is important to enter the `{$}` brackets as well, since this tells the processor that the image URL is contained in an XSL variable called $watermark.



6. Select **Apply Layout**.
7. Save the document and test your new watermark.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent nibh nulla, porta ut, suscipit ac, luctus vitae, eros. Pellentesque nisl. Mauris et ipsum. Donec condimentum eros porta nulla. Mauris ac dui. Cras euismod nunc. Morbi enim sem, volutpat eu, porta consectetur, cursus nec est. Cras facilisis ipsum sit amet ipsum. Aliquam justo leo, laoreet et, sagittis id, aliquam vel, massa. Donec pharetra arcu scelerisque sapien. Maecenas pretium feugiat pede. Aliquam erat volutpat. Pellentesque tempus luctus libero. Mauris nibh felis, ornare id, rhoncus sit amet, convallis et, libero. Praesent tristique eleifend pede. Sed solicitudin ipsum ut est. Donec a felis a nunc iaculis laoreet. Ut id justo.

Sed euismod, eros quis lacinia vestibulum, sem metus blandit lectus, ut fringilla lectus odio vitae mi. Nunc in leo. Fusce non augue. Nunc tempus sem ac pede. Sed enim orci imperdiet in, congue posuere, aliquam id, neque. Nam condimentum. Proin adipiscing mattis enim. Duis sed sapien. Donec at tortor mattis ante faucibus eleifend. Pellentesque vel ipsum a dui rutrum aliquam. Donec laoreet pharetra elit. Suspendisse vitae odio. Donec a augue id urna lobortis volutpat. Nulla eleifend. Vivamus ut enim.

Vivamus leo urna, facilisis sed, accumsan nec, volutpat id, ante. Praesent laoreet facilisis magna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In velit. Morbi vel nisi. Phasellus dui orci, tincidunt at, ornare ut, tristique sed, lorem. Donec feugiat. Aliquam quam. Nulla porttitor fermentum diam. Donec ornare, nulla ut adipiscing accumsan, ante mi posuere arcu, sagittis posuere urna elit at velit. Ut diam. Curabitur nec turpis a enim euismod laoreet. Pellentesque pulvinar, risus id laoreet adipiscing, magna massa elementum est, non aliquam quam nisl eget orci. Donec ante est, eleifend sed, semper tempus, ultricies id, lectus. Proin dapibus mauris feugiat turpis. Morbi vel nibh.

Sed neque augue, consequat et, convallis sit amet, facilisis non, enim. Pellentesque pulvinar felis eget mauris. In hac habitasse platea dictumst. Suspendisse semper eleifend nulla. Nunc consequat. Morbi purus. Pellentesque a

The watermark image will only appear in the document output when printed and will not be displayed inside of the page editor.

NOTES

Documents are deployed separately from commerce. To view the latest changes in a document on a Print/Email/Preview action, you must first deploy the document. New documents must be deployed before they can be linked or accessed in the Printer-Friendly XSL views.

Doc Engine now includes the capability to format all BigMachines-supported currencies (as long as the symbol is supported by the font).

Only the deployed version of a document is available for download or migration. Global elements, like Content Library and Complex Conditionals, will be migrated every time a document is migrated.

Bulk upload of an existing document using the "Add" action is not supported.

RELATED TOPICS

DOCUMENT ENGINE OVERVIEW

OVERVIEW

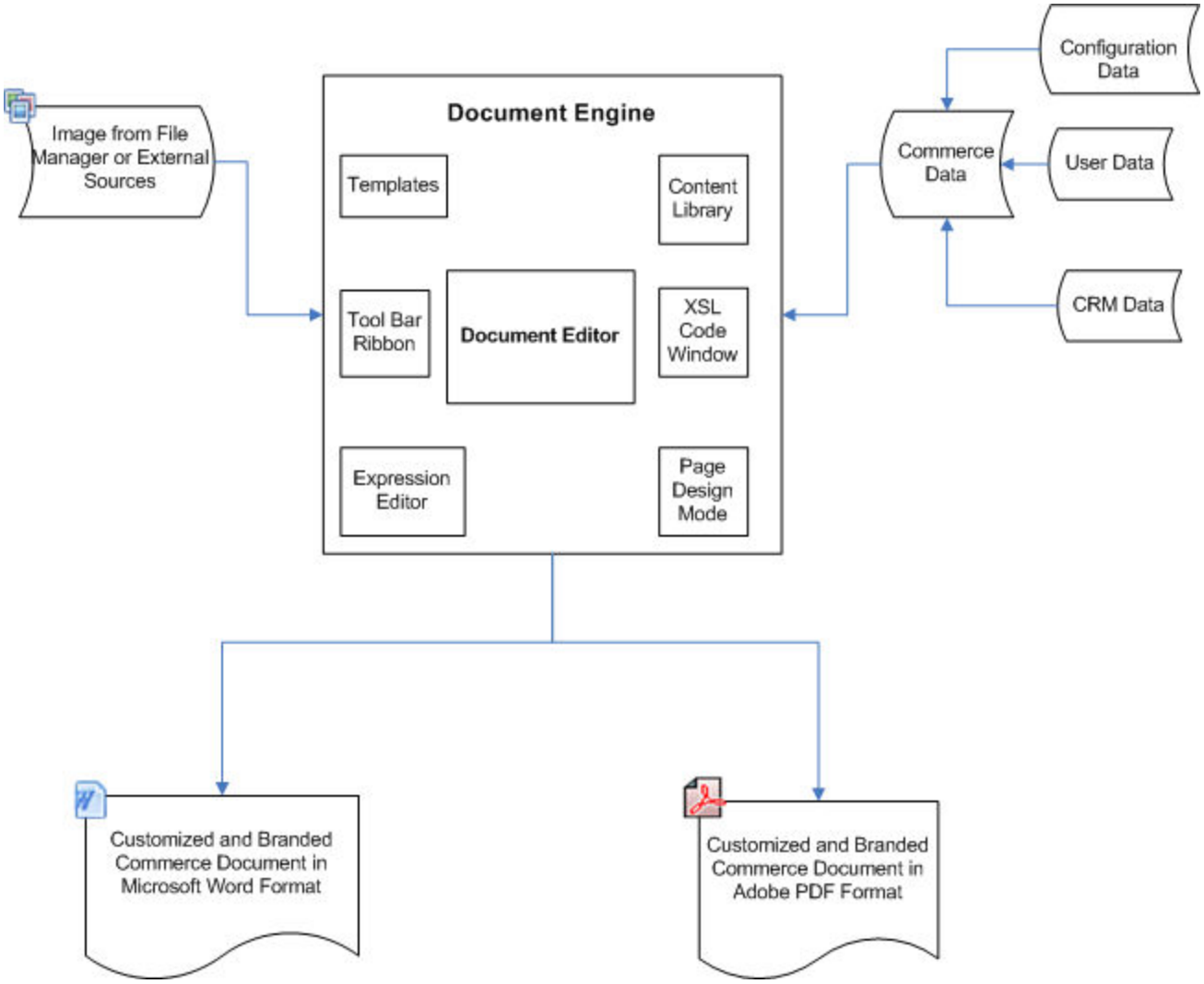
The Document Engine powers the BigMachines Proposal Generator and the BigMachines Contract Generator. Users can create professional and customer-friendly proposals and submittals as well as customize and brand proposals for specific products, channel partners or customers. It also automates the generation of customized contracts driven from approved and submitted quotes and orders.

Document customization capabilities:

- Populate proposals with relevant customer data
- Include detailed quotes as sub-documents or appended-documents
- Add customized terms and conditions based on configuration options
- Add customized and branded cover letters from user-defined and/or company-defined templates
- Add technical data sheets, pictures, and drawings
- Add quality documentation or industry specifications
- Browse and append any external documents from the user's desktop, company intranet, or internal shared drives
- Generate proposals, quotes or contracts in MS Word or Adobe PDF

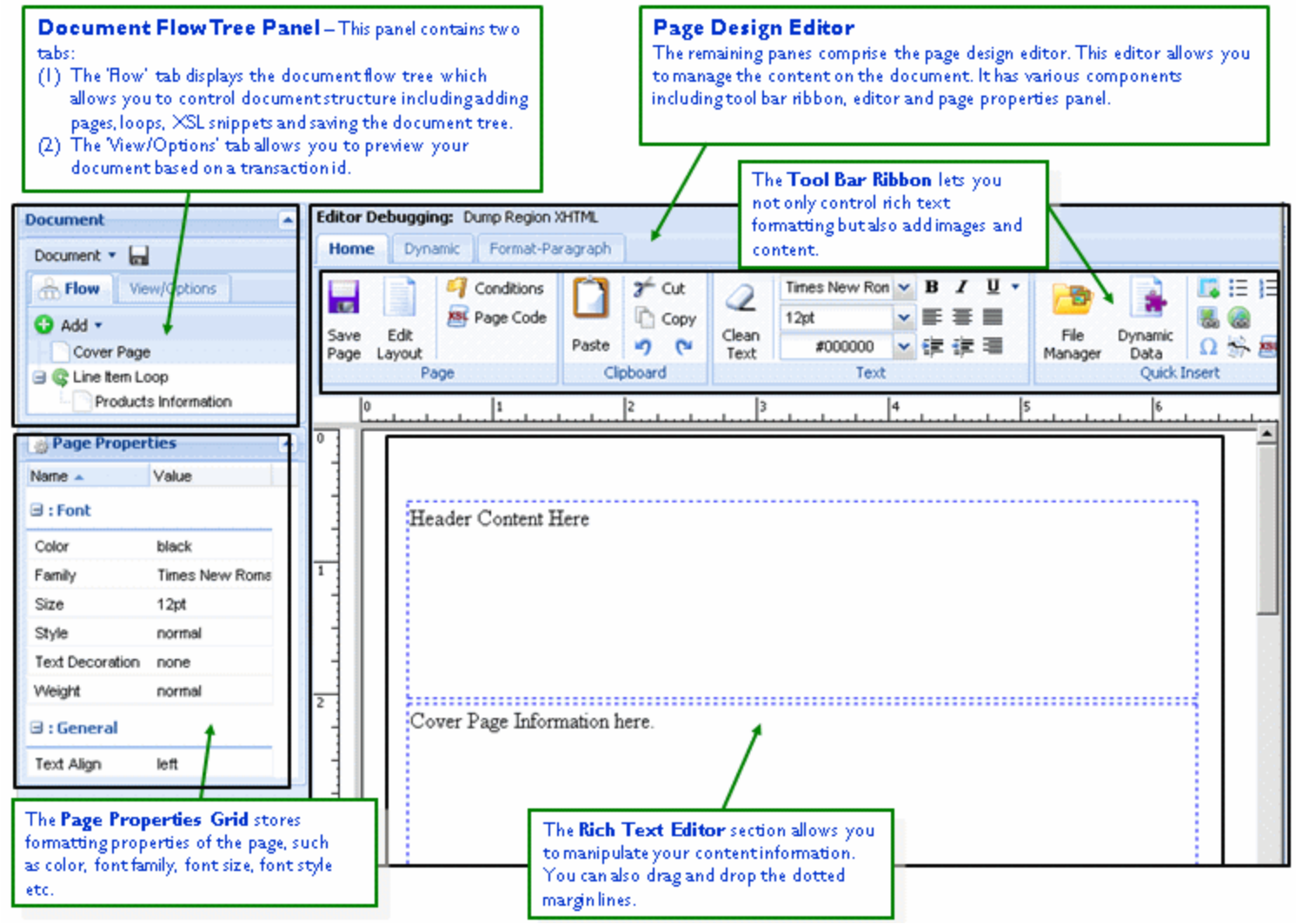
ADMINISTRATION

Document Engine Structure & Relationships



Document Editor

The BigMachines Document Editor is a visual XSL editor designed for authoring and editing XSL templates. Users can produce dynamic and customized commerce documents in either PDF or RTF formats. This visual XSL Editor helps combine the ease of a WYSIWYG tool with the power of XSL, facilitating quick and error-free template development. The document editor contains various components and features that make it so powerful and easy to use. Using a combination of the components and the features, you should be able to put together a custom proposal very quickly.



Key Components of Document Editor

Pages

In the Document Engine, a Page does not represent a literal page as your body content may end up flowing over to a new page when it's printed, depending on the data on the user-side. Think of a Document Engine Page as a place where you can define a constant header and/or footer with fixed dimensions. If your body content overflows to more pages, those pages will share the header/footer that you defined on the Document Engine Page. Additionally, you can insert hard page breaks into your page body content. In technical terms, a Page will output a complete `xsl<fo:page -sequence>` node.

Loops

Loops in the Document Flow Tree repeat over all line items and can display individual Product Information pages for each line item in the loop. You can edit the loop to control how the pages within a loop are generated. In technical terms, a loop in an `<xsl:-for-each>` which will ultimately be inserted outside of the <fo:page-sequence> tags that are generated for each Document Flow Page. Thus Document Flow Loops are useful for creating repeating pages.

XSL Snippets

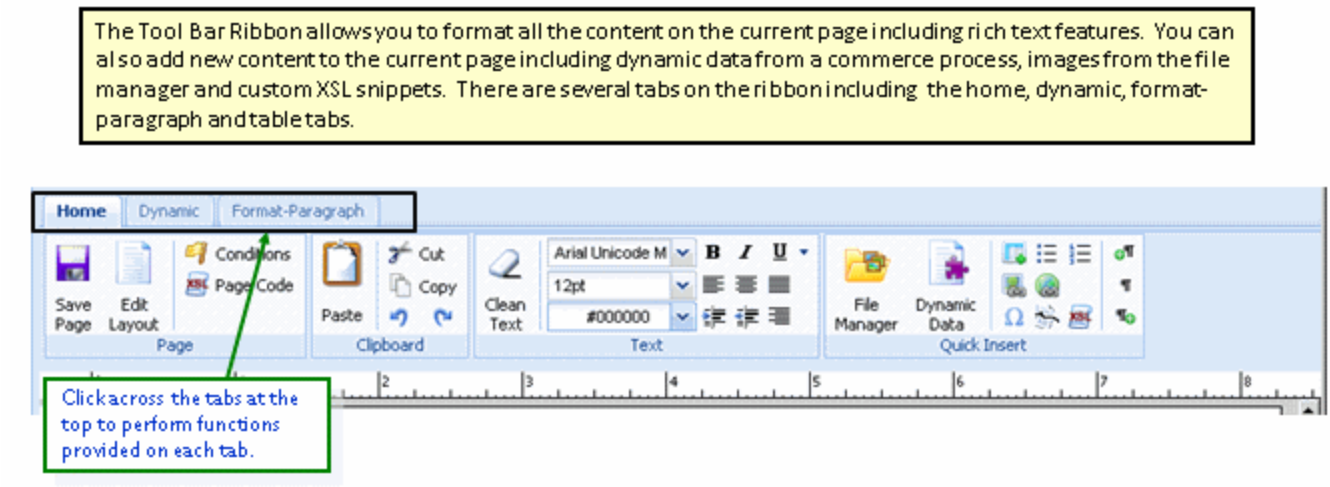
The code that you write in a Document Flow XSL Snippet will be inserted between the `<fo:page-sequence>` nodes that are generated by Page objects. Therefore, to avoid XSL errors, you should only write XSL that will either be `<fo:page-sequence>` nodes or generate `<fo:page-sequence>` nodes.

Features in a Document

- Enhanced Grid View section to view and edit the page properties
- Easy-to-use and rich user interface for Page Editing
    - Tabbed-ribbon tool bar
    - Page regions and margins can be set through dragging
- Access to the file manager directly from the Editor
    - Insert images and links from the file manager through drag-and-drop
    - Add and manager files using the User Interface
- New formatting features such as
- Nested lists, custom bullets
- Paste from Microsoft Word retains formatting
- Table Formatting
- Preview and debug the document to test, perfect and optimize the template before converting it
- At any time during the XSL template editing process, the user can transform the file(s) with a single click to preview the generated output as either PDF, Advanced PDF or RTF
- Edit the Global XSL snippet and maintain global variables, templates, etc.
- Customize the existing editor XSL data (such as <fo:simple-page-master>) in an advanced XSL code panel
- Quick and Easy Book Marking and Table of Contents

Tool Bar Ribbon

The Tool Bar contains actions that the user can perform on the elements of a Document Page in the Page Design Mode. The actions on the Ribbon are grouped in tabs. CliSelectingcking on a tab switches the tool bar actions to a different set of actions. For more information, see Toolbar.



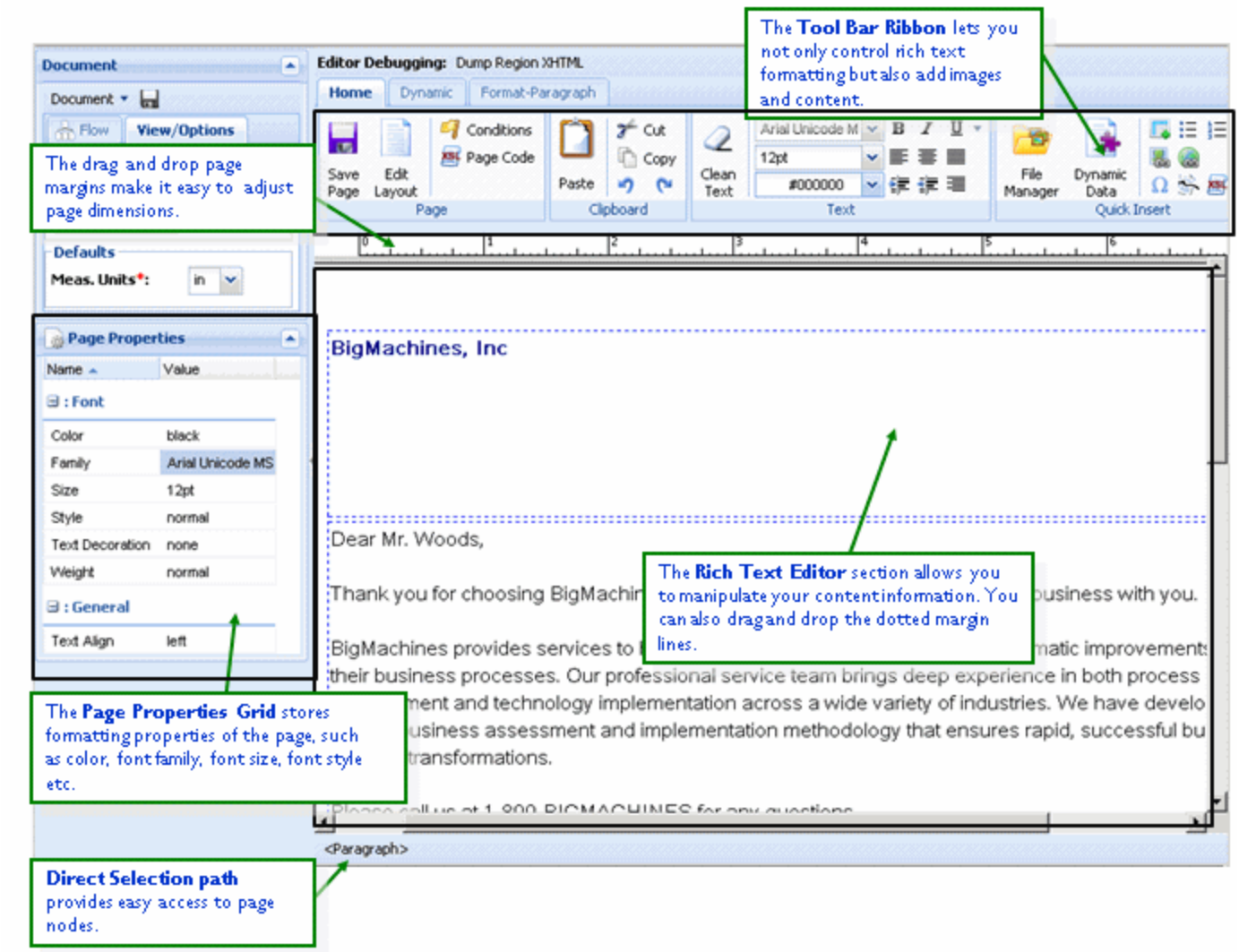
The visibility of tabs on the ribbon depends on the current active page region. **Example:** Table tab is only visible when the cursor is inside a table. In addition, tool bar items within a tab can be enabled or disabled based on whether it is appropriate for the current selection. **Example:** borders are only applicable to block nodes like paragraphs, tables, etc. and not for in line elements. In this case, the border actions will be disabled if the user is currently editing an in line selection.

Available Tabs on the Tool Bar Ribbon

- **Home** - This is the default tab, which contains the most common actions. The Home tab is always available in the Page Design Mode. See Home Tab Actions.
- **Block Format** - The Block Format tab is always available in the Page Design Mode. The actions on this tab allow the user to format the selected node.
- **Dynamic** - The Dynamic tab includes actions that allow the user to customize their pages.
- **Table** - The Table tab allows the user to customize tables in a page. This tab is visible only when the selected node is a table.
- **Image** - The Image tab is visible only when the selected node is an image. The actions in this tab can be used to change the source URL and the height and width of the image. Since images have fixed widths, the user must specify the image size.
- **Hyper Link** - The Hyper Link tab is visible only when the selected node is a hyper link. You can use this tab to edit the hyper link.
- **Book Mark** -The Book Mark tab allows the user to change the name of the bookmark. This tab is visible only when the selected node is a bookmark. A bookmark name must be unique to a page.
- **List** - The List tab allows the user to customize lists in a page. This tab is visible only when the selected node is a list or a list element. You can create both ordered and unordered lists. BigMachines also allows you to create nested lists.

Page Design Mode

The Page Design Mode is where the user builds and edits Pages. This mode includes a collection of components for authoring and formatting text. The user can also insert dynamic data in the text, which is data that easily and automatically changes to allow for easy customization. Navigation: Document Application Panel > Document Flow Tree > desired page



Components of Page Design Mode

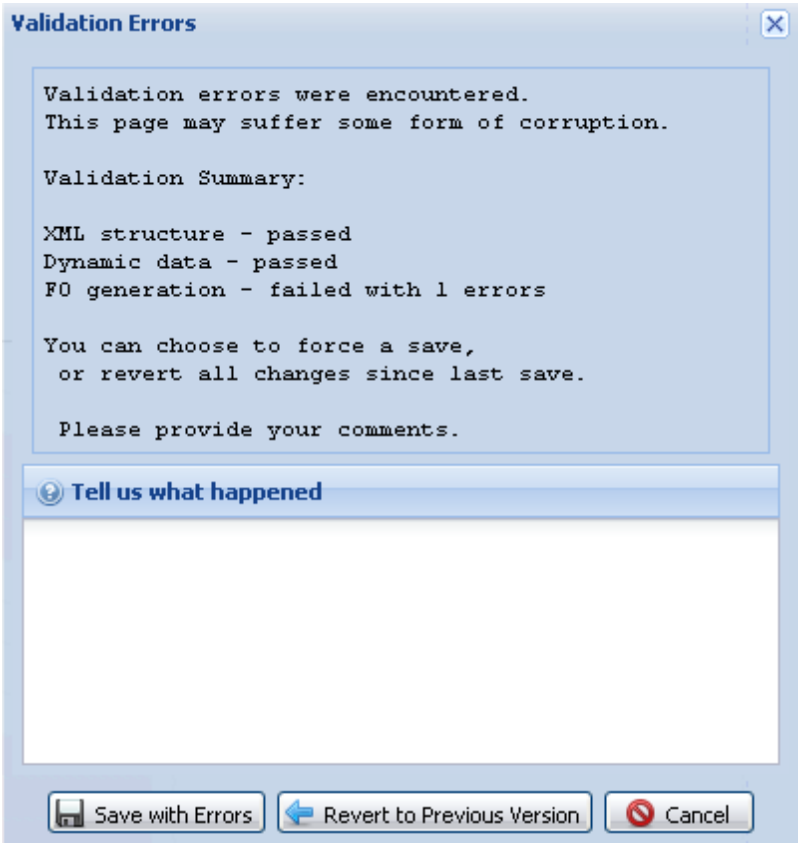
- **Page Properties Grid:** The grid displayed on the left panel stores formatting properties of the page, such as color, font family, font size, font style etc.
- **Page Design Tool Bar Ribbon:** The Tool Bar Ribbon comprises a menu of actions that can be used to edit and format the document pages. The tool bar consists of a set of tabs with groups of actions. Selecting a specific tab switches the tool bar to display the chosen tab with its menu items.
- **Rich Text Editor Region:** The user can view and edit the text within a page. The user can also choose to enter dynamic data using the Dynamic Content Window. The action on the Ribbon can be used to manipulate the content in this region.

- **Page Vertical & Horizontal Rulers:** The vertical and horizontal rulers provide a sense of the document's dimension for easier formatting. The rulers are defaulted to the Document Default Units option that can be changed in the Settings Tab of the Document Application Panel.
- **Draggable Page Layout/Margins:** The dimensions of the pages can be changed easily by dragging on the margins.
- **Direct Selection Path:** The user can select entire nodes in the RTE. This is useful in that the user can select the exact node to editing or formatting. Selecting a node button selects the entire node in the Rich Text Editor.

Validations & Optional Feedback

You can optionally run a validation on page data when saving in the Document Engine. When an error is found during validation, you will get an error message. This will allow you to revert to the last saved copy, cancel the save or save the corruption. You can send feedback to the BigMachines development team directly when you get the corruption. *Make sure to define the steps to reproduce.*

1. Navigate from: Admin Homepage > Commerce and Documents > Commerce Settings.
2. Select Yes for the option **Document Engine – Validate Page Data on Save**. *This is set to NO by default.*



Upon validation error, you can

- **Save with Errors:** Save despite any errors. From this point on, you will encounter this error on this content until it is fixed.
- **Revert to Previous Version:** Revert to the last saved state. This will not rollback changes made to content of dynamic items like XSL snippets, simple conditions, loops, clause conditions, row conditions/loops, formatted expressions and others.
- **Cancel:** Cancel the operation which caused this validation to fire. You will not lose changes you made.



The validation is run on complex conditionals.

The error message is sent to development, even if you don't provide feedback details. This feedback is NOT replacing immediate bug reporting, so turn around time for this would be much longer. Feedback is stored in a database that is periodically analyzed by development.



Revert to last save is not available for new complex conditionals. *This will be available once the complex conditional has been saved for the first time.*

Supported Node Types

Content in the Rich Text Editor is organized in a way similar to the way HTML is structured. This not only helps in providing structure to pages in a document, but also allows for enhanced functionality. **Example:** depending on the node type that is selected, specific ribbon tabs can be shown/hidden. This lets the user know exactly what actions can be performed on a given node type. The Structural Building Blocks of a document are known as Nodes.

Supported Node Type	Description
P	paragraph
Table	table
TR	row in a table
TD	cell in a table row
SPAN	inline style
IMG	image
A	anchor link
UL	list
Li	list item
LOOP	repeating block

Supported Node Type	Description
CONDBLOCK	condition block (if...then)
IFELSE	if else
DATA	dynamic data
BMK	bookmark
PGNUM	page number



Some Node Types share names with corresponding HTML tags with similar functionality. However, these are custom nodes are not to be confused with HTML tags. Node are strings, and each can be translated.

Best Practices

- While working in the Doc Engine Editor, click the  button to save your work.
- Use standard functionality over XSL when possible.
- Make backups.
- After modifying a document page through FireBug and saving the changes, reload/reopen the page before making any further changes. *This ensures that the html changes made using FireBug are properly reflected in the document XSL.*

TROUBLESHOOTING

Fixing corrupt documents

1. Corrupted file error messages are sometimes caused by deleting/altering tables, rows, or conditionals. *Follow these steps to rectify the situation:*
2. Open the corrupted file in Doc Engine.
3. Navigate to the area where an item was recently deleted/edited prior to the file corruption.
4. Open Firebug within Doc Engine.
5. Click the “Click an element in the page to inspect" button in the upper left hand corner.
6. Click the row (or table, paragraph w/ conditional) that was recently altered and may have caused the corruption. *The element you selected will be highlighted in the Firebug editor.*
7. Examine the element's HTML to determine if it is causing the problem.

8. Once the corrupted element has been located, right-click the element and select “Delete Element.” *You will need to rebuild the table (or row) that had been there.*



When deleting items in Doc Engine, begin with the innermost element (e.g. dynamic data, XSL snippets, complex conditionals) and then work outwards (e.g. row, table).



To avoid this problem in the future, delete elements properly by using the buttons provided within Doc Engine.

USE CASES

Use Case: Displaying a Conditional Watermark

Let's say you need to display a background "DRAFT" image for quotes that are in a pending state. We recommended that you define a global `main_doc` variable to point to your header attributes. You can only define it once in the document, so if `main_doc` is already defined, don't define it again.

1. Upload an image to the BigMachines File Manager to use as the watermark. *You will need to do this first in the BigMachines applications since you can't upload files through the Document Engine. Make sure that the image that you use is a valid image that the FO processor can read.*

2. Add the following to define a custom watermark image:

    - Replace the image URL to point to your watermark image.

    - Change the logic of the text to one that you need for your application. In the example below, we are checking for the main document commerce attribute with variable name `quoteStatus_q` and seeing if it equals the text "Pending". *Your implementation may be different. You can add more <xsl:when> statements if you need to.*

    - An `<xsl:choose>` statement is basically an `IF/ELSE IF/ELSE` block. *The processor will retrieve the value of the first <xsl:when> conditional statement that is true. In this case, we are returning the URL of the watermark if the watermark should be displayed.*

```
<xsl:variable name="main_doc" select="/transaction/data_xml/document[@data_type=0]"/>

<xsl:variable name="watermark">

<xsl:choose>

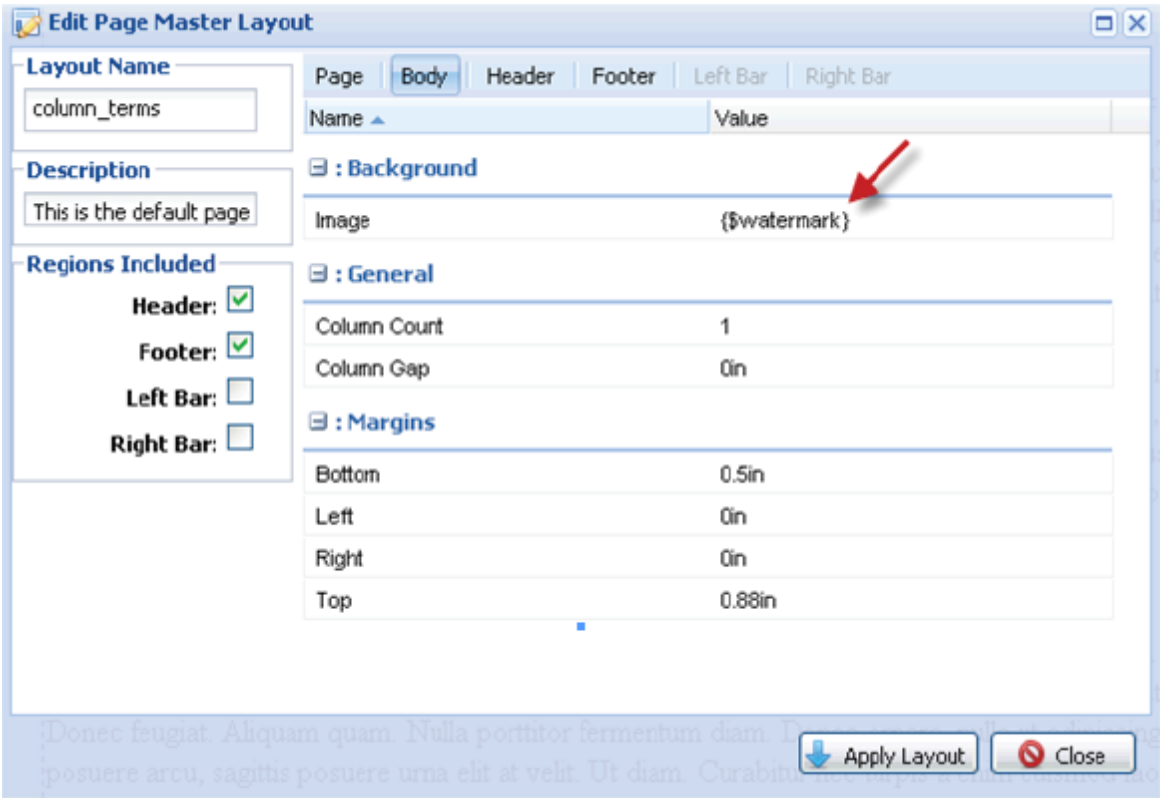
<xsl:when test="$main_doc/quoteStatus_q = 'Pend-
ing">/bmfsweb/pm/image/Proposal/watermark.jpg</xsl:when>

<xsl:otherwise></xsl:otherwise>

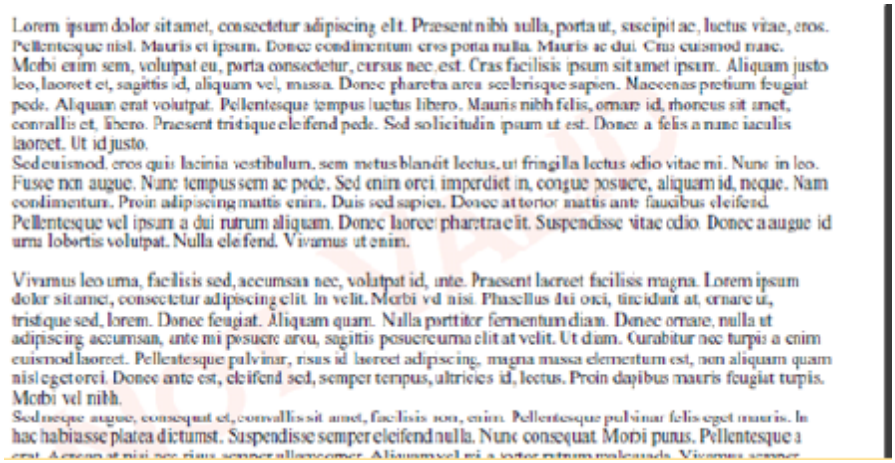
</xsl:choose>
```

```
</xsl:variable>
```

3. Save your **Global Snippet** AND save the document.
4. Open the page where you want to place the watermark. Select **Edit Layout** from the Home Tool bar to open the Layout Editor.
5. Click on the **Body** tab and go to the image property. *You will tell this page layout to read the global watermark variable.* Enter `{$watermark}` into the Image field. It is important to enter the `{$}` brackets as well, since this tells the processor that the image URL is contained in an XSL variable called $watermark.



6. Select **Apply Layout**.
7. Save the document and test your new watermark.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent nibh nulla, porta ut, suscipit ac, luctus vitae, eros. Pellentesque nisl. Mauris et ipsum. Donec condimentum eros porta nulla. Mauris ac dui. Cras euismod nunc. Morbi enim sem, volutpat eu, porta consectetur, cursus nec est. Cras facilisis ipsum sit amet ipsum. Aliquam justo leo, laoreet et, sagittis id, aliquam vel, massa. Donec pharetra arcu scelerisque sapien. Maecenas pretium feugiat pede. Aliquam erat volutpat. Pellentesque tempus luctus libero. Mauris nibh felis, ornare id, rhoncus sit amet, convallis et, libero. Praesent tristique eleifend pede. Sed solicitudin ipsum ut est. Donec a felis a nunc iaculis laoreet. Ut id justo.

Sed euismod, eros quis lacinia vestibulum, sem metus blandit lectus, ut fringilla lectus odio vitae mi. Nunc in leo. Fusce non augue. Nunc tempus sem ac pede. Sed enim orci imperdiet in, congue posuere, aliquam id, neque. Nam condimentum. Proin adipiscing mattis enim. Duis sed sapien. Donec at tortor mattis ante faucibus eleifend. Pellentesque vel ipsum a dui rutrum aliquam. Donec laoreet pharetra elit. Suspendisse vitae odio. Donec a augue id urna lobortis volutpat. Nulla eleifend. Vivamus ut enim.

Vivamus leo urna, facilisis sed, accumsan nec, volutpat id, ante. Praesent laoreet facilisis magna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In velit. Morbi vel nisi. Phasellus dui orci, tincidunt at, ornare ut, tristique sed, lorem. Donec feugiat. Aliquam quam. Nulla porttitor fermentum diam. Donec ornare, nulla ut adipiscing accumsan, ante mi posuere arcu, sagittis posuere urna elit at velit. Ut diam. Curabitur nec turpis a enim euismod laoreet. Pellentesque pulvinar, risus id laoreet adipiscing, magna massa elementum est, non aliquam quam nisl eget orci. Donec ante est, eleifend sed, semper tempus, ultricies id, lectus. Proin dapibus mauris feugiat turpis. Morbi vel nibh.

Sed neque augue, consequat et, convallis sit amet, facilisis non, enim. Pellentesque pulvinar felis eget mauris. In hac habitasse platea dictumst. Suspendisse semper eleifend nulla. Nunc consequat. Morbi purus. Pellentesque a

The watermark image will only appear in the document output when printed and will not be displayed inside of the page editor.

NOTES

Documents are deployed separately from commerce. To view the latest changes in a document on a Print/Email/Preview action, you must first deploy the document. New documents must be deployed before they can be linked or accessed in the Printer-Friendly XSL views.

Doc Engine now includes the capability to format all BigMachines-supported currencies (as long as the symbol is supported by the font).

Only the deployed version of a document is available for download or migration. Global elements, like Content Library and Complex Conditionals, will be migrated every time a document is migrated.

Bulk upload of an existing document using the "Add" action is not supported.

RELATED TOPICS

DOCUMENT ENGINE MANAGEMENT CENTER

OVERVIEW

From Documents, you can manage your documents as well as which Printer-Friendly XSL Views they will be attached to. In the left-most panel, you'll see the header Processes.

Navigation: Admin Platform > Commerce and Documents > Documents



In the right-most panel, you'll see the header Document Editor List. This is where you will perform the majority of your actions.



Common Processes

Icon	Process
	Clicking on the XSL icon will open a pop-up window where you can view the XSL of the Complex Conditionals that were created and associated with this commerce process.
	Clicking on this icon will refresh your data sources for the commerce process. Note: This can also be done within the Document Engine and should be done anytime changes are made to the commerce process. This will ensure that the most up-to-date information is available within the Doc Engine. Remember that the commerce process must be deployed for these changes to take effect.

Specialized Actions

Icon	Action
	Clicking this icon next to the desired document will allow you to view the last deployed document XSL.
	Clicking the pencil icon will allow you to edit the name and description of the document.
	Clicking the copy icon will make an exact copy, or a clone, of the selected document. Note: An un-deployed document cannot be copied.

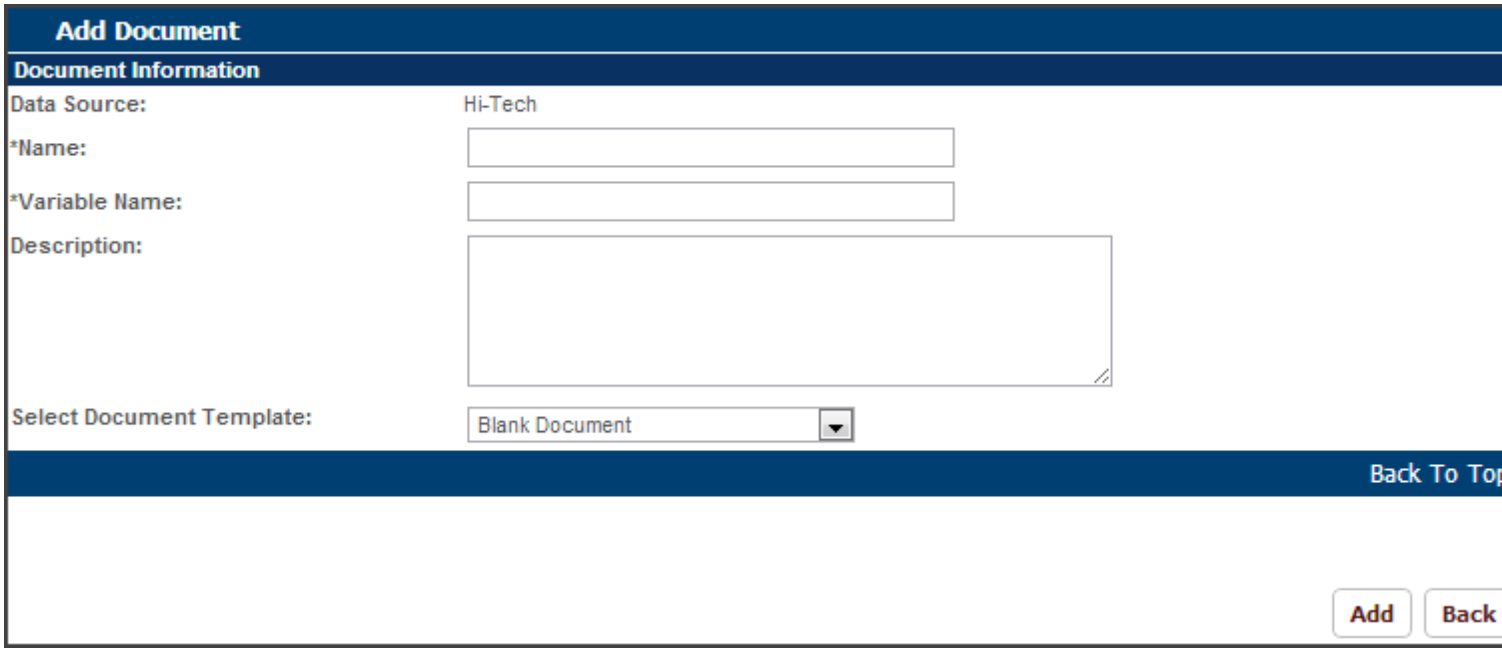
Icon	Action
	Clicking the export icon will allow you to export the document in XSL and upload it to a different site or process.
	Clicking the deploy icon for a selected document will allow you to deploy documents one at a time. Un-deployed documents will be shown in red.
List	Clicking the list icon under References will show you which Printer-Friendly XSL's the document is attached to. You can click on the XSL view name to open the printer-friendly XSL view editor.

ADMINISTRATION

Almost of the document engine administration can now be completed within Documents.

Creating a new document

1. From the Document Editor List, select **Add Document**.



2. Enter the **Name**, **Variable Name** and **Description** for your document. *Optional: Select the document template. The default selection is Blank Document.*
3. Click **Add**.
4. Select the **Open Document Editor** button to open the Doc Engine and begin creating your document.

5. Once your have finished editing your document, select **Deploy**.



To edit your Documents, click on the Document Name and the Document Editor will pop-up.

Importing a new document

1. From the Document Editor list, located the document that you'd like to migrate and select the **Export** icon.



The export ZIP has the same name as the document that is being exported. It should be noted that special characters in the file name will cause the subsequent upload to fail. The allowed characters are alpha_numeric and underscore. You are advised to rename the ZIP file if there are any special characters in the document name.

2. Save the ZIP file to your computer.

3. Select the **Import** button the Document Editor List page of the production site, or commerce process, that you are migrating the document to.



Documents on the test site and the production site are not considered equivalent if both are manually created, even if they have the same variable name. For documents to be equivalent in production and test, the production documents must have been created through an upload action OR the two sties must be exact database copies.

Next Steps: Once you have created or migrated documents to the Document List, you'll need to link them to a printer-friendly XSL view.

NOTES



Create or update documents on your test site and then migrate them to production.

Documents are deployed separately from commerce. To view the latest changes in a document you'll need to deploy the document.

Changes to documents are visible to the end user without having to deploy the commerce process.



New documents must be deployed before they can be accessed in the Printer-Friendly XSL view.

Only deployed versions of documents are available for download or migration. *Undeployed changes from the test site will not be available in the production site after migrating or uploading documents.*

Global elements like the Content Library and Complex Conditionals will be migrated every time the document is migrated.



Bulk upload of an existing document using the Add action is not supported.

RELATED TOPICS

CONTENT LIBRARY

OVERVIEW

The Content Library stores formatted dynamic text,loops,conditionals, and XSL Code Snippets. The users can add document elements to the library and later insert these elements anywhere in the document as well as other documents in the commerce process.
Users can save the following content categories in the Library tab:

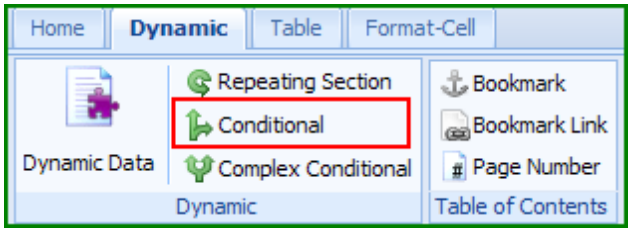
- Loop Conditions and Content
- If/Else Conditional Blocks and Content
- Formatted Dynamic Data Expressions
- XSL Code Snippets - XSL Code Snippets help to organize and use the code in an efficient manner

ADMINISTRATION

Adding items to the Content Library

Use Case: You want to save a conditional section of terms & conditions so it can be used by other documents in the commerce process.

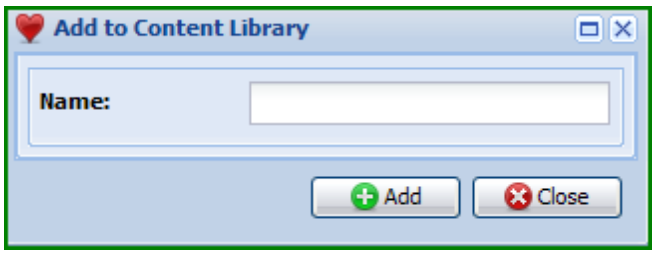
1. Select the **Conditional** button on the **Dynamic** toolbar tab.



2. Select the heart icon next to the element to be added. *This opens the Add to Content Library dialog.*

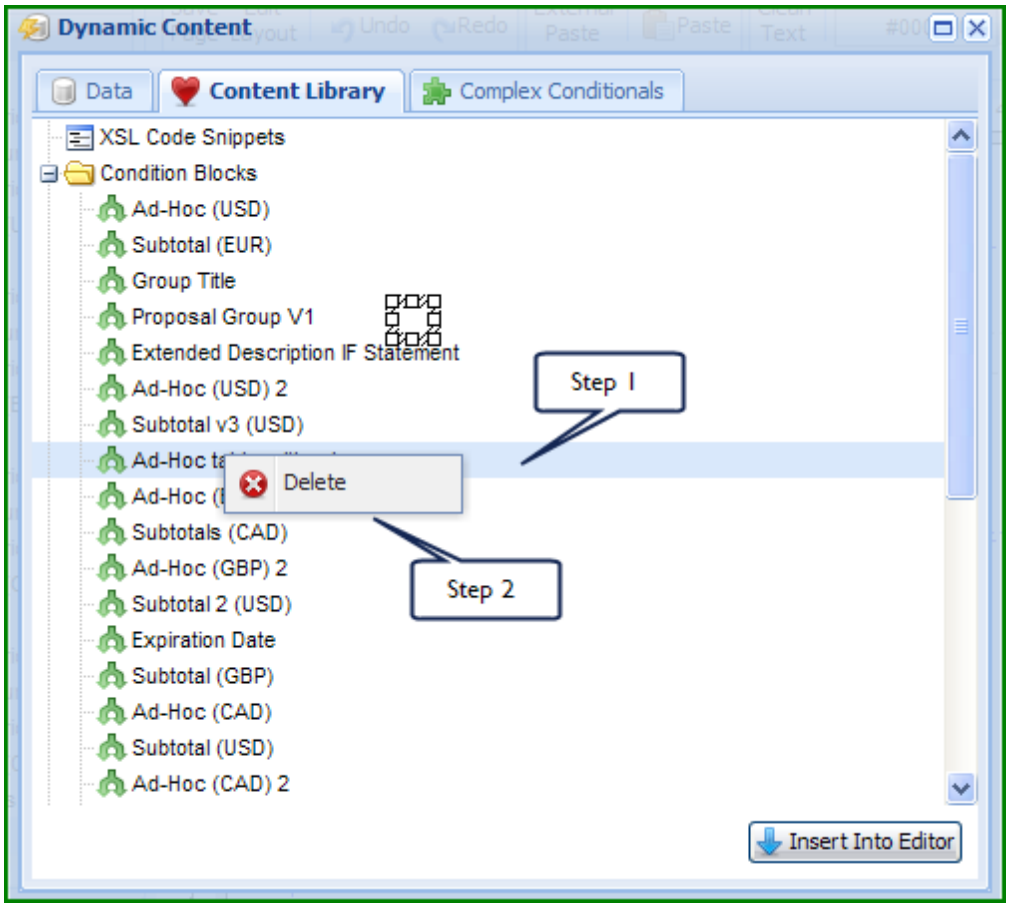


3. Enter your conditional **Name** and select **Add**. *After the content library tree refreshes itself, you will see your new item displayed under the appropriate folder.*



Removing items from the Content Library

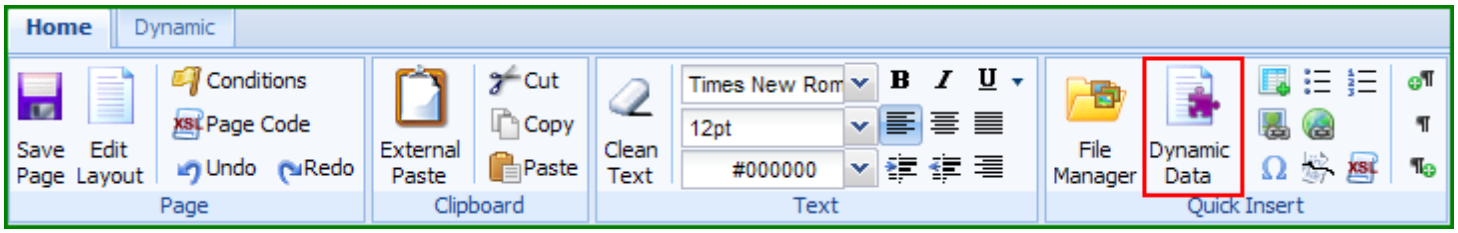
1. From the Content Library tree, right click on the desired item to delete it.
2. Select **Delete** from the menu.



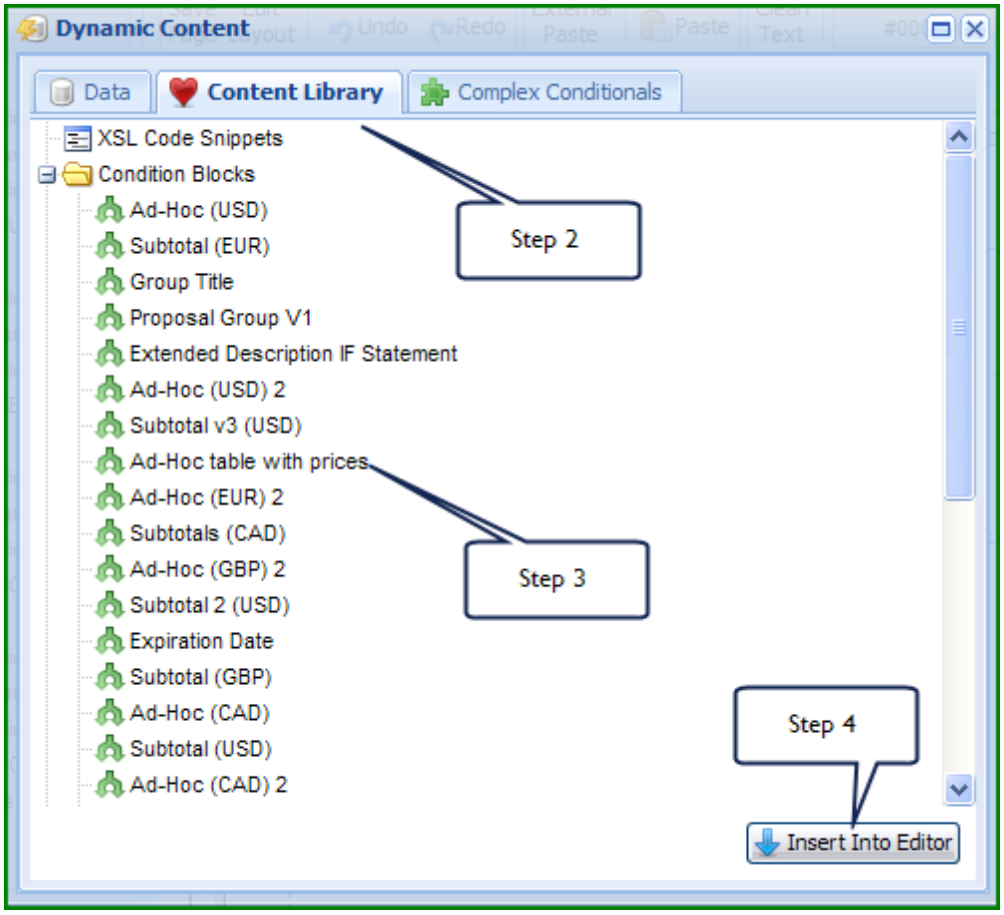
This will remove the item from the tree, but will NOT remove any inserted content in the editor.

Inserting content from the Content Library

1. Open the Dynamic Data dialog by selecting **Dynamic Data** on the Home or Dynamic toolbar tabs.



2. In the Dynamic Content dialog, select the **Content Library** tab.



3. Place your cursor at a place in the editor where you want to insert the content and locate the desired item in the **Content Library** tree.

4. Select **Insert Into Editor**. *This drops the desired content on the node that has been selected.*



Items added to the library are automatically stored in pre-defined folders depending on the type of content. The users cannot move or re-order these items within the folder. They also cannot move these items into different folders.

NOTES



When content is added to the Content Library, the copy maintained in the Library can no longer be edited. This means that if the user adds some content to the library and then changes the item later, the content maintained in the library does not change. *To get the changes, the user must re-add the item to the library and drop the newly added item at his desired node.*



Any content that is inserted by the Content Library does not contain any references to the source item.

Any changes that you make to the content that you insert will NOT update the item in the Content Library or any other insertions of the same content.

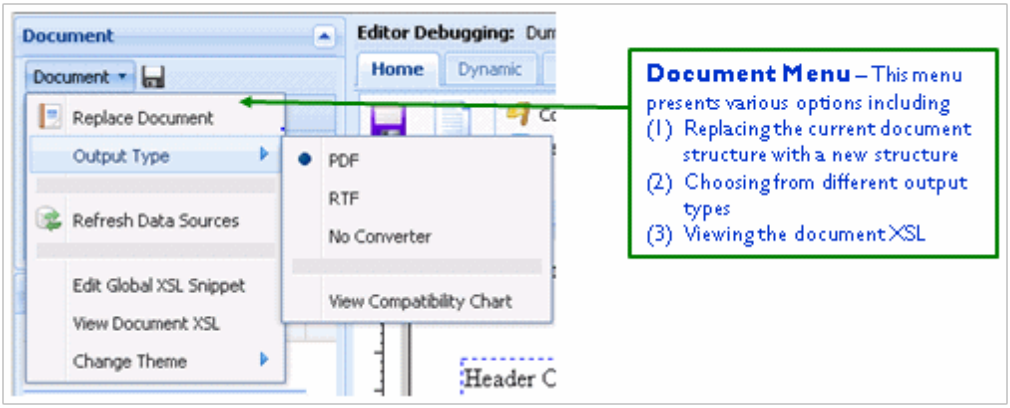
RELATED TOPICS

DOCUMENT PANEL

OVERVIEW

The Document Panel is the first panel that the user sees when the BigMachines Document Editor is launched. This panel controls the structure of the current document that may contain pages, loops, and XSL snippets. This section also allows the user to preview the output and set default units. It is also used to specify the document settings. **Major components of the Document Panel:**

Document Menu



Document menu drop-down options:

- **Replace Document:** This lets you replace the entire contents of your current document tree structure with new documents.
- **Output Type:** This helps the user to change the Output Type. The output type that the end user sees is determined by the Process Output that is selected on the Add/Update XSL View page. However, this menu item allows the user to preview the output as either RTF, PDF or Advanced PDF. *Changing from one output type to another in an existing document is discouraged as this might introduce comparability issues.*
- **Refresh Data Source:** This action refreshes the external data that is used to generate the document. For example, this makes new transaction data available to the proposal editor that might impact the dynamic text embedded in the pages. The user must click on this action after every test to production.
- **Edit Global XSL Snippet:** This action allows the users to edit XSL code that can be maintained outside the global scope of the document (outside of the main `<xsl:tempate-math="/">`). The users can use this snippet to maintain global variables and templates. *By default, this template is empty.*
- **View Document XSL:** This action allows the users to view the complete generated XSL of the document that has been created with the Document Editor.



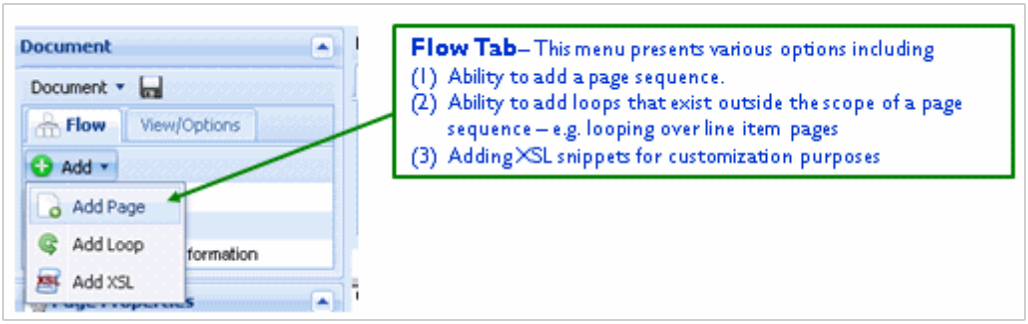
You can save the Document XSL to your local machine in order to make changes with an XSL editor. The edited XSL can then be uploaded as a Custom XSL. Changes made within the Document XSL Code window is not saved. To view the

XSL for an individual page, navigate to the page and click on the Page Code action on the Home Tab.

- **Change Theme:** This allows you to change the skin on the entire document editor window.

Flow Tab

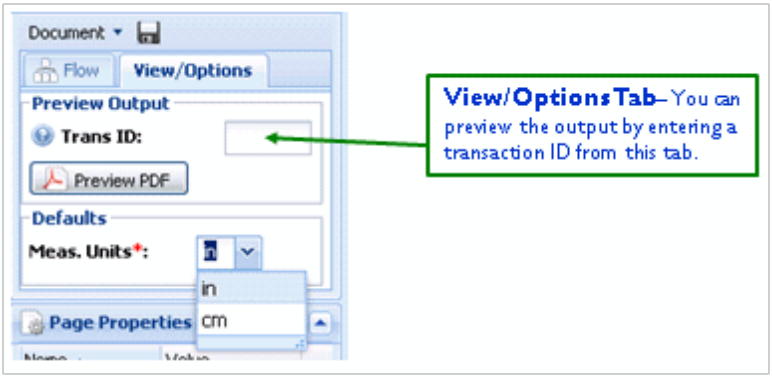
The Flow Tab allows the users to add, deleted and rename document elements. The elements of a document are organized in a hierarchical tree to visually represent how the document is laid out. The users have to click on the name of an element to edit it.



Document Elements	Description
Page	Does not represent a literal page as your body content may end up flowing over to a new page, depending on the data on the user-side when the document is printed. Think of a Document Engine Page as a place where you can define a constant header and/or footer with fixed dimensions. If your body content overflows to more pages, those pages will share the header/footer that you defined on the Document Engine Page. Additionally, you can insert hard page breaks into your page body content. In technical terms, a Page will output a complete xsl `<fo:pagesequence>` node.
Loop	Loops in the Document Flow Tree repeat over all the line items and can display individual Products Information pages for each line item in the loop. You can edit the loop to control how the pages within a loop are generated. In technical terms, a loop is an `<xsl:for-each>` which will ultimately be inserted outside of the `<fo:page-sequence>` tags that are generated for each Document Flow Page. Thus, Document Flow Loops are useful for creating repeating pages.
XSL Snippet	The code that you write in a Document Flow XSL Snippet will be inserted between the `<fo:pagesequence>` nodes that are generated by Page objects. Therefore, to avoid XSL errors, you should only write XSL that will either be or generate `<fo:page-sequence>` nodes.

View/Options Tab

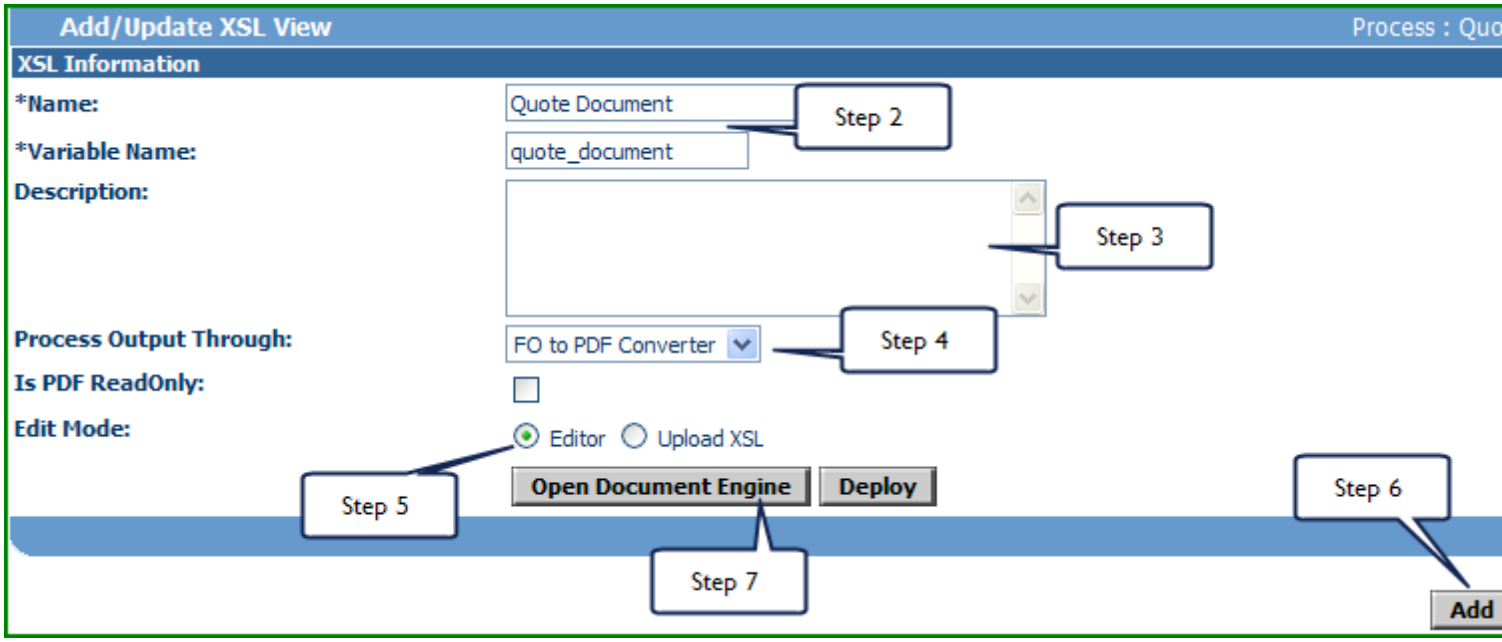
This tab allows the user to preview the output in the PDF or RTF format. The output type is determined by the Process Output that is selected on the Add/Update XSL View page. However, the user can change the output type for preview by choosing a different output type in the Document Panel.



ADMINISTRATION

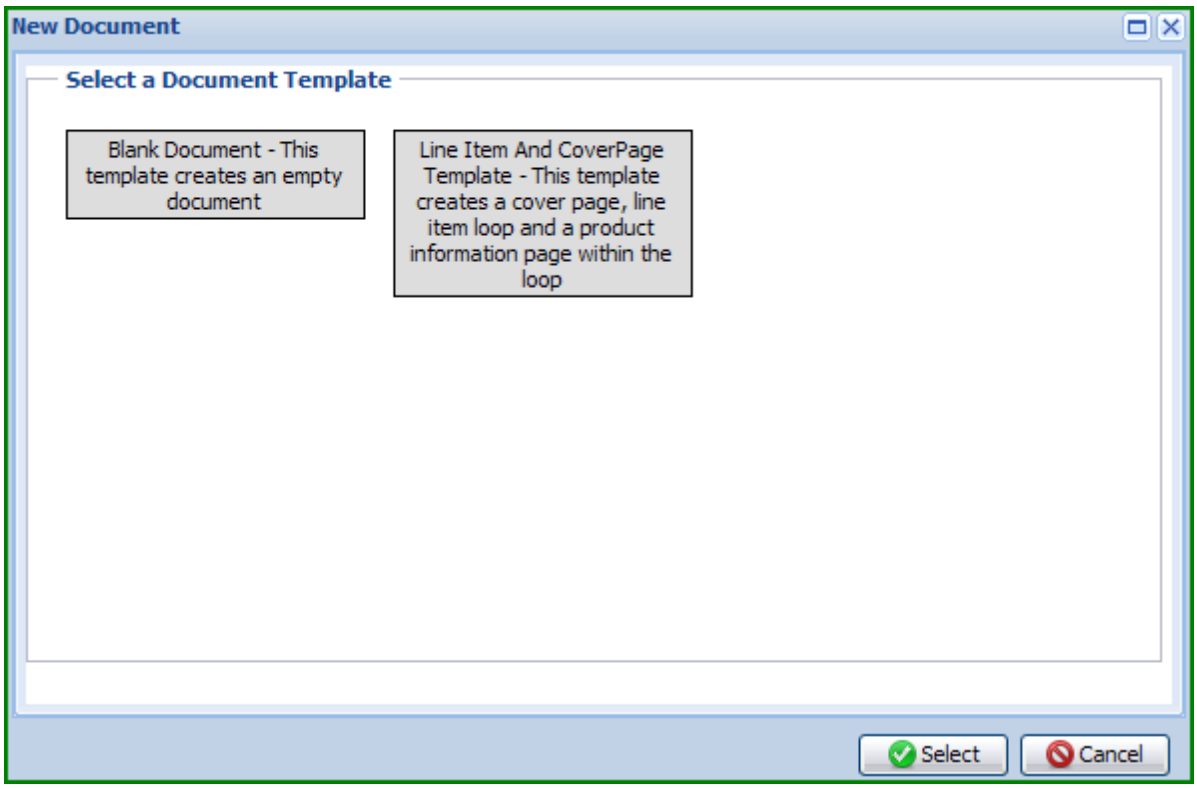
Creating a New Document

1. Click **Add** on the Printer Friend/History XSL Views list page.
2. Enter a **Name** and **Variable Name** for the document.
3. Enter a description for the document.
4. Choose a **Process Output Through** from the drop-down menu. The options are: "No Converter", "FO to RTF Converter", "FO to PDF Converter". *If you select "FO to PDF Converter", determine if the output will be ReadOnly. If so, check the Is PDF ReadOnly box.*



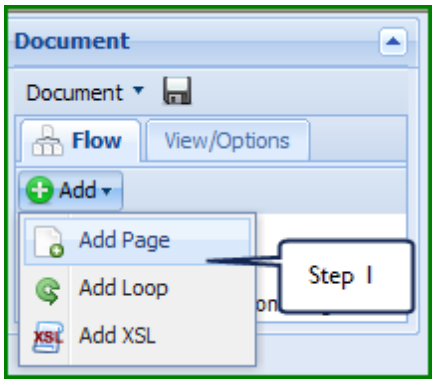
5. Click **Editor** as the edit mode.
6. Click **Add**.

7. Select **Open Document Engine**.
    - *The Document Engine will open in a new window where you can select a document template from which to start. You will have the following choices:*
        - The Blank Document template creates a blank document with no predefined pages.
        - The Line Item and Cover Page template creates a Cover Page Document followed by Product Information pages for each line item. A line item loop ensures that individual pages are created for each line item in a document.
8. Locate your desired template and click **Select**.

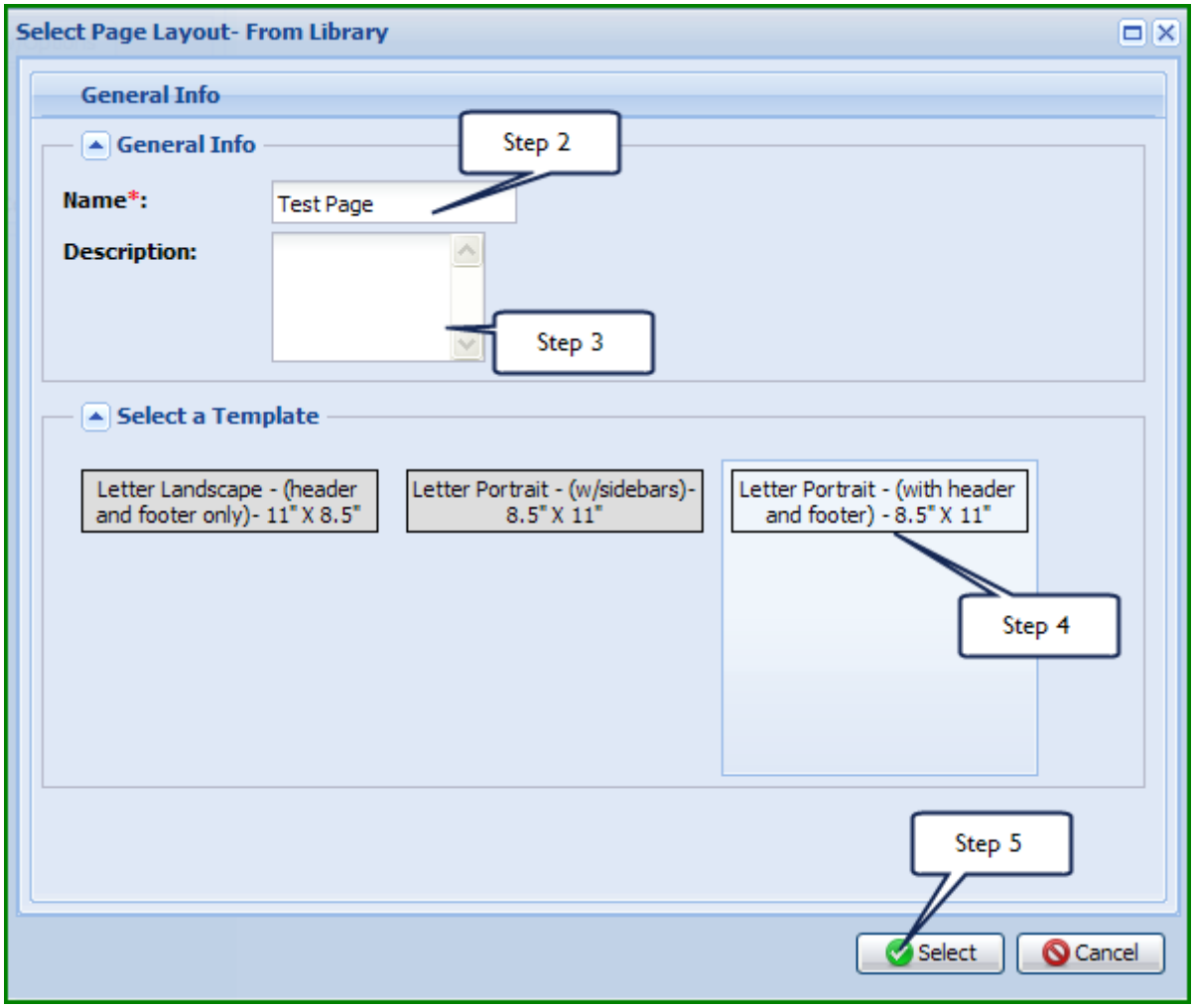


Adding a New Page

1. Choose **Add Page** from the Add drop-down menu on the Flow Tab.



2. Enter a name into the **Page Layout Library**. *A dialog box will open.*
3. Enter a valid **Description** for your new page.
4. Select from a list of page layout templates to start from. *In most cases, the first option **Default Page** will suffice.*



5. Click **Select** to finalize.

By default, your new page will be appended to the end of the Document Flow Tree.

You can drag and drop the Page to its desired location in the flow. You can rename your Page by right-clicking on it and selecting "Rename" from the menu.

Adding a Loop

You will add a loop when you want a page to print out for each line item in your transaction, sorted by their sequence number.

1. Select **Add Loop** from the Document Panel's Flow Tab.
2. Define properties within the Loop Expression Editor dialog that opens.
3. Click **Save**.



A new loop object will appear in the flow tree.

NOTES



You can save the Document XSL to your local machine in order to make changes with an XSL editor. The edited XSL can then be uploaded as a Custom XSL. Changes made within the Document XSL Code window is not saved. To view the XSL for an individual page, navigate to the page and click on the Page Code action on the Home Tab.



Changing from one output type to another is discouraged as this might introduce compatibility issues.

RELATED TOPICS

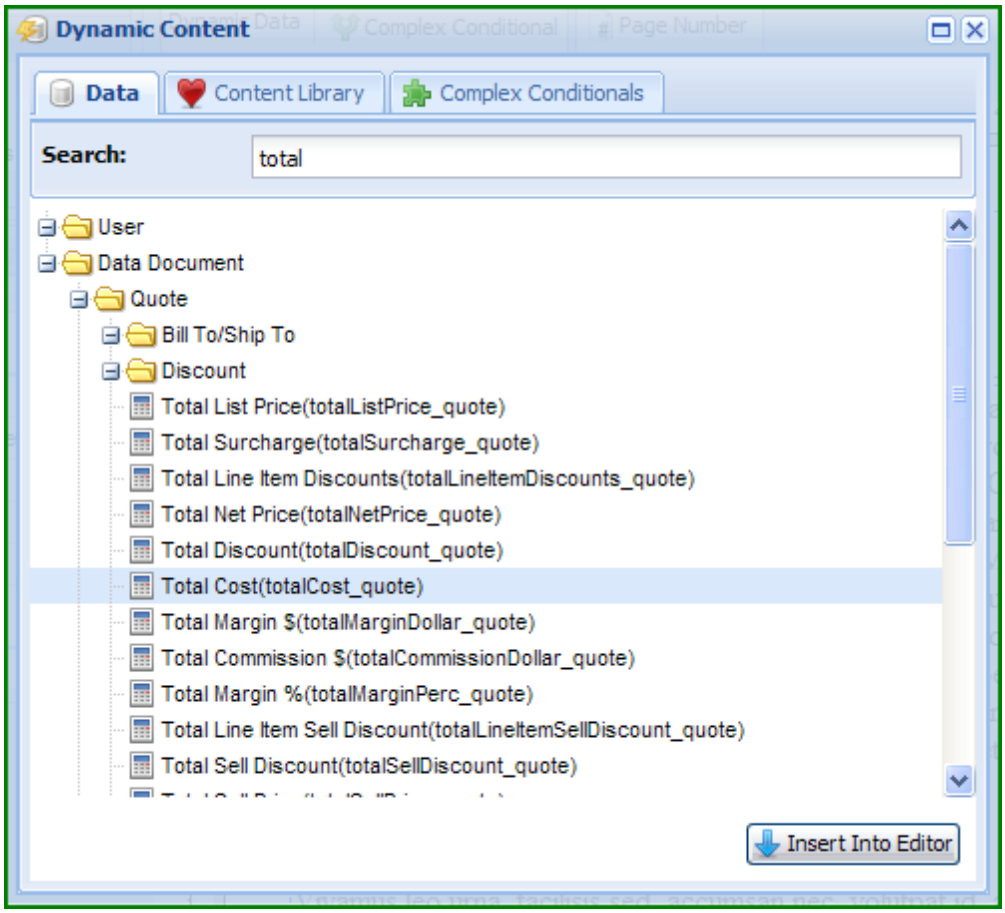
DYNAMIC CONTENT WINDOW

OVERVIEW

Navigation: Document Editor > Home Tab > Dynamic Data button **OR** Document Editor > Dynamic Tab > Dynamic Data button

Tabs available on the Document Content Window:

- **Data:** This is the default tab displayed on this window that displays the transaction XML files as a tree.
- **Content Library:** The Library tab enables the user to save blocks of text for re-use in other places in the document.
- **Complex Conditionals:** The Complex Conditionals Library allows the user to save and use complex condition statements with multiple if...else clauses inline within the content of a document using the Document Editor without having to resort to custom XSL snippets.



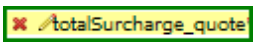
ADMINISTRATION

Inserting Dynamic Data into Editor

1. Search by filtering the data you want to display by entering text in the Search Field. *This will automatically display only the specific attributes/nodes that you want to view.*

2. Select the Search Source. *This is a read-only field that indicates the current commerce process, which is the source for all the dynamic data.*

3. To insert data, you can either: drag and drop the attribute, double-click on the attribute **OR** click the **Insert Into Editor** button.



   - *Dragging the content into the editor, highlights the background color of the specific region. Drop the content into the editor section.*
   - *The inserted data text is highlighted in yellow and contains two action buttons next to it (delete and edit).*

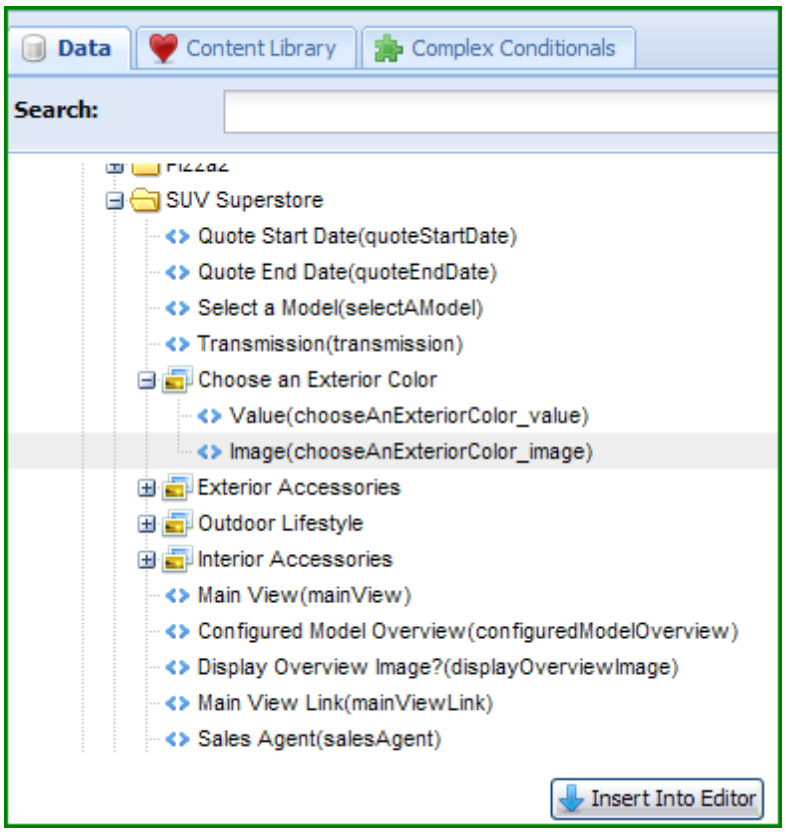
Inserting Library Content into Editor

1. Drag and drop the items from the tree into the editor section. *Dragging the content into the editor, highlights the background color of the specific region.*

2. Drop the content into the editor section. *The inserted data text is highlighted in yellow and contains two action buttons next to it.*

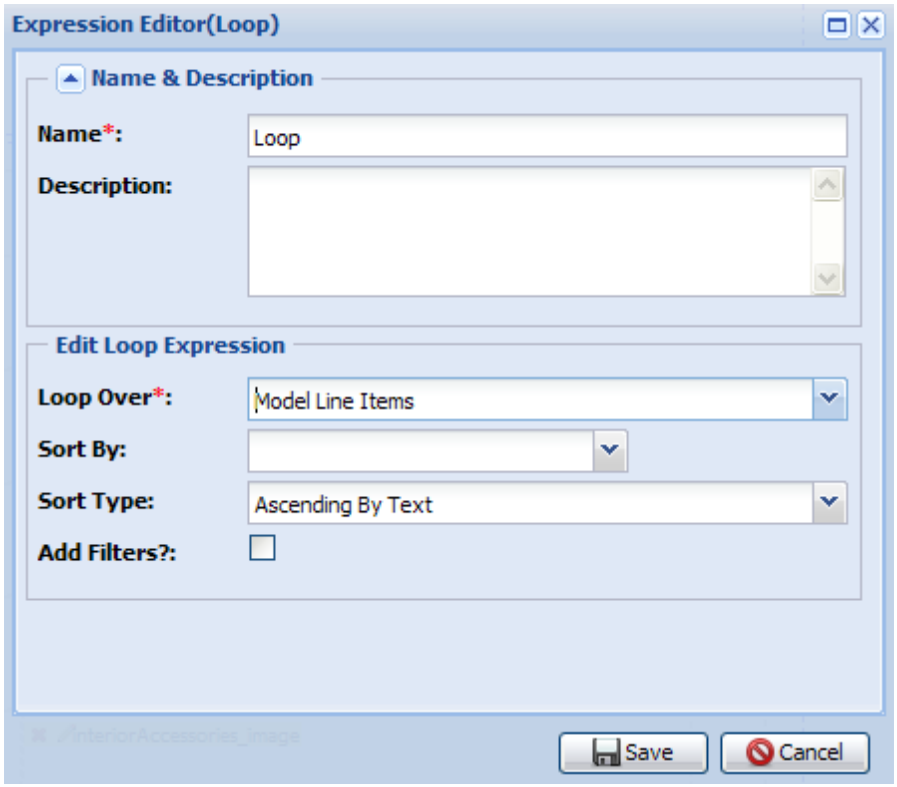
Image Menus in Document Engine

In the Data Tab, each menu now has two tags: Value and Image. The fullaccessuser can choose to display the image of a Visual Menu in the printed document, the text value or both. *Notice the "picture" icon next to the configurable attributes that contain image menus.*



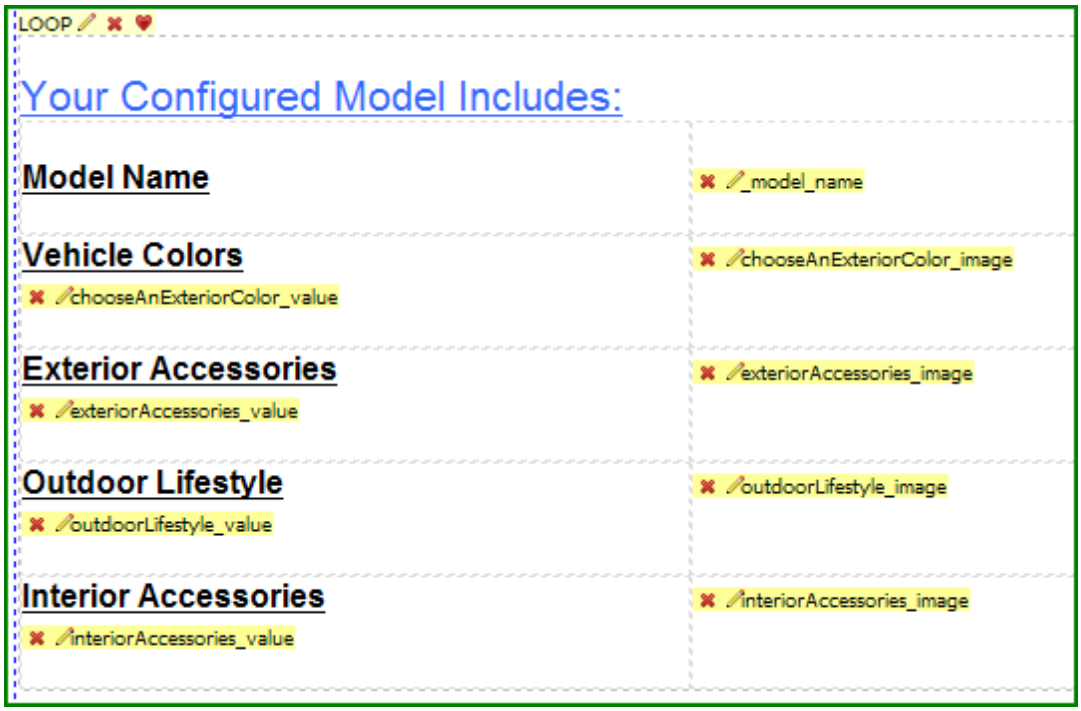
Adding image menu values to Document Engine

1. Place your cursor on the page where you would like to display the text image values of your configurable attribute.
2. Select the Dynamic Tab.
3. Select the **Repeating Section** icon to display the Loop Expression Editor dialog.
4. Choose **Model Line Items** from the **Loop Over** drop-down menu. *You can sort and set filters if you would like to.*



5. Click **Save**.
6. Within the loop, enter your configurable attribute content. *Best practice is to set your information within a table. As we*

introduced at the beginning of this section, you can choose to enter the text or image value from the Data Tab.



Example of your new loop within your printer-friendly document:

Your Configured Model Includes:

Model Name Mid Sized SUV

Vehicle Colors
Satin White Pearl



Exterior Accessories
Front Underspoiler



Outdoor Lifestyle
Roof Cargo Carrier,Ski Attachment





Interior Accessories
Ash Tray,Auto Dim Mirror





If the parent node of an image menu is inserted into the document, only the text value will be printed. Images are displayed only when the image specific tag is inserted into the document.

NOTES



The Image Menu section for printing simple multi-select menus and arrays will print the values in a row.

You will need to create a custom XSL or use a table to loop through your menus to achieve better formatting.



Make sure that all of your images are either JPG or non-transparent GIF ONLY. Failure to do so can results in an XSL error message: "Error in processing data. Please modify XSL file".

RELATED TOPICS

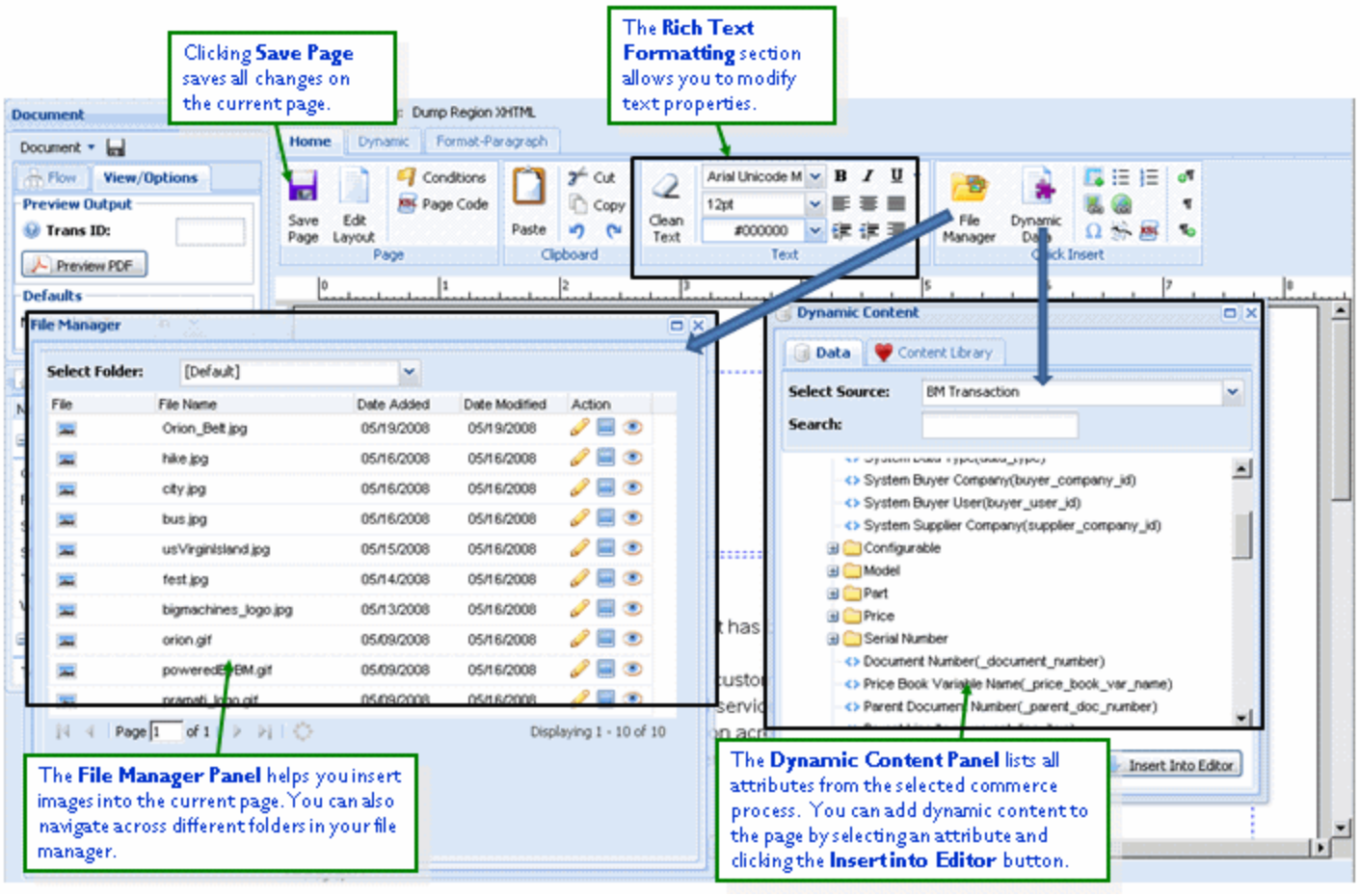
TOOLBAR

OVERVIEW

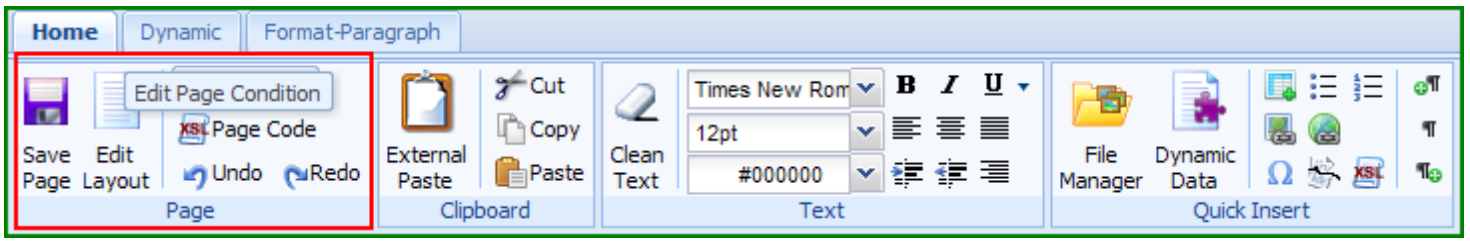
When creating or editing a document within Document Engine, you will use the toolbar to add dynamic data, format text, and create/edit tables.

Home Tab

The Home Tab is the default tab, and contains the most commonly used actions. These actions are stored within groups.



Page Related Actions

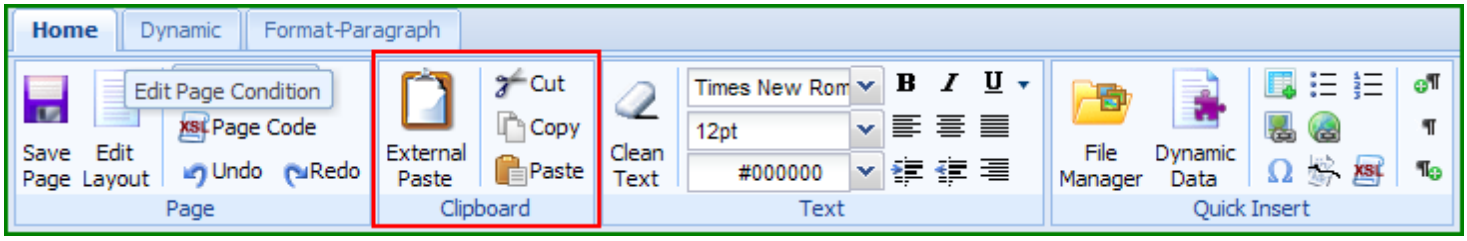


- **Save Page:** Saves the current page along with the associated layout.
- **Edit Layout:** Click this button to open a Page Master Layout Editor window. The user can use this window to specify/edit the layout dimensions of the active page.
- **Conditions:** Use this option to open the Conditional Expression Editor window to define/edit page conditions.
- **Page Code:** Use this button to bring up the edit XSL Code page. This page contains the XSL code generated for the page open in the editor section
- **Page Design:** Use this button to return to the Page Design page.

> If you edit the XSL and then click on the Page Design action, the changes are not saved. However, if you click on Save the XSL Code button, the user cannot return to the page design mode for this page. The code is then saved as an XSL snippet and can be edited within the Document Editor.

- **Undo:** Use this button to undo the last active editor action.
- **Redo:** Use this button to undo the last active editor action.

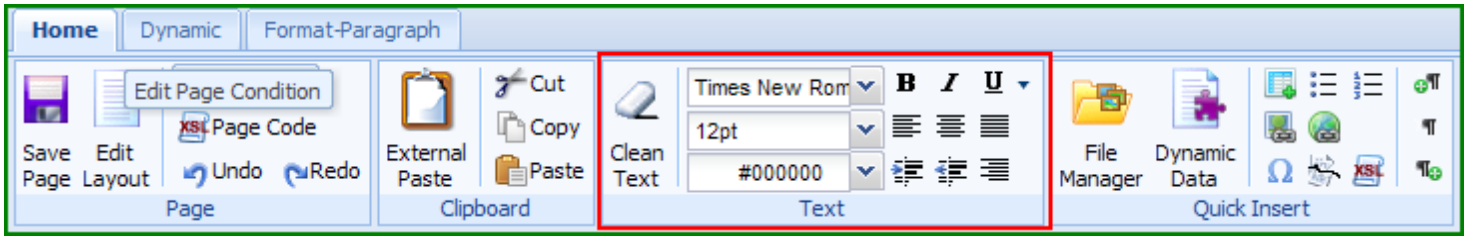
Clipboard Related Actions



- **External Paste:** Allows you to copy from an external application like Word, Notepad, etc.
- **Cut:** Use this to cut a selection of text and store it in the clipboard.
- **Copy:** Use this to copy a selection of text and store it in the clipboard.
- **Paste:** Use this option to paste the clipboard content into the editor. *This is typically used in conjunction with Cut or Copy. Pasting content into the editor cleans up any formatting automatically before inserting it into the editor.*

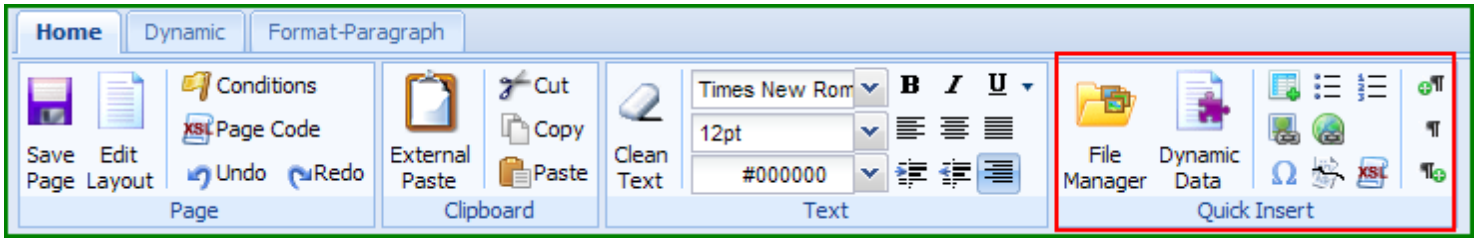
Text Related Actions

This group comprises of a set of formatting characteristics that you can apply to text, tables, and lists in your templates to quickly change their appearance.



- **Clean Text:** Use this option to remove the current format of the text and restore the formatting to the normal style for your content. This includes classes and styles related to formatting.
- **Font Family:** Select a font-family from the drop-down menu. The menu contains all supported-fonts.
- **Font Size:** Select a font size using the drop-down menu or type in a custom size. All font sizes are defined in points (pt).
- **Font Color:** Enter a hex color or select a color from the drop-down menu.
- **Font Formatting:**
  - **Bold:** Select text and click the bold icon.
  - **Italic:** Select text and click the italic icon.
  - **Underline:** Select text and click the underline icon.
  - **Underline Sub-Set:** To access additional underline options, select the drop-down menu next to the Underline icon.
    - **Strike-through:** *Select the text to apply this formatting to and click Strike-through to place a line through the selected text.*
    - **Superscript:** *Select the text that you need to modify and click superscript to apply formatting.*
    - **Subscript:** *Select the text that needs to be modified and click subscript to apply formatting.*
- **Alignment:** Select Align Left, Align Center, Justify Full and Align Right.
- **Indent/Out-dent:** Select the text to indent or out-dent or place your cursor on the line that needs to be indented/out-dented.

Quick Insert Related Actions



- **File Manager:** Clicking on this opens the File Manager window. *Files cannot be added, deleted, or replaced from the Document Engine.*
- **Dynamic Data:** Insert dynamic data.
- **Insert Table:** Insert, create, and edit tables.
- **Insert Unordered List:** Use this button to insert a bulleted list into the editing area. *The lists can be multi-nested.*
- **Insert Ordered List:** Use this button to insert a numbered list. By default, the numbers scheme is a simple decimal format (1, 2, 3, etc.). *After a list has been inserted, the list properties (bullet, etc.) can be edited through the List tab. Lists can be nested in as many levels as needed.*
- **Insert Image from External URL:** Use this option to insert an image from an external URL. The Text field always displays the direct URL to the file (this includes the `HTTP://` for external images).
- **Insert Hyperlink:** Use this option to insert a hyperlink.
- **Insert Special Character:** Use this option to open a window where the user can select from a visual list of characters to insert into the rich text editor. *For the special symbol to show up on the printed document, the user must choose the Helvetica font for special symbols.*
- **Insert Page Break:** Insert a manual page break and start a new page.
- **Insert XSL:** Insert XSL code into your document.

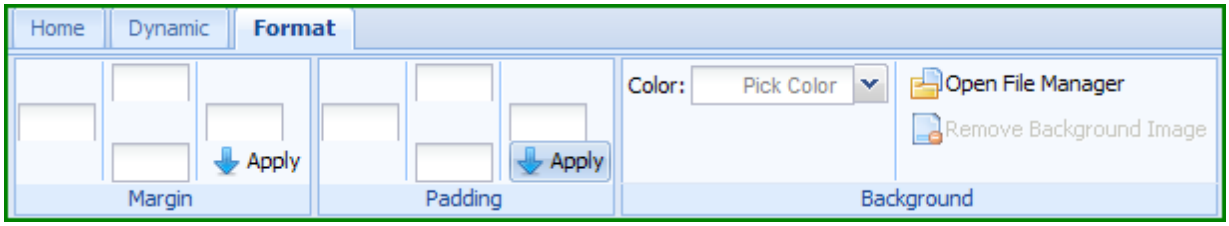


Both PDF and RTF outputs will accept JPG and GIF image files, however GIF files CANNOT be transparent.

RTF files can also accept PNG files.

Format Tab

You can format elements in a document such as lists, paragraphs, and templates. This tab is always visible to the user. The name of this tab changes based on the current selected node indicate that the formatting on this tab applies to the select node only. To specify format for the whole page, the user has to go to the Page Properties panel.



Action groups & related Actions:

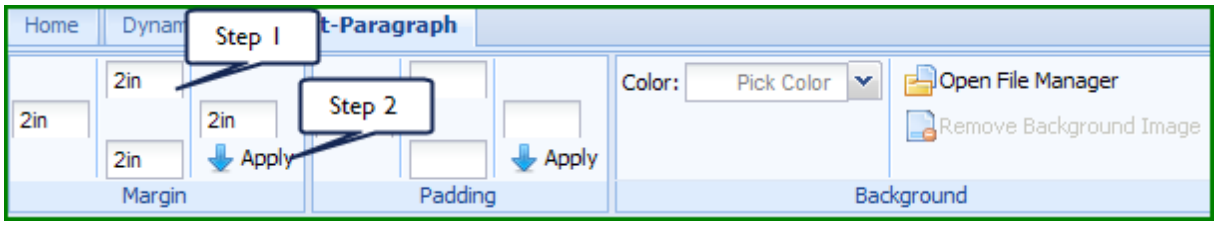
- **Margins Group:** This group specifies the Margin Size of the chosen Node. *Margins can be defined on the four sides - Top, Bottom, Left, and Right.*
- **Padding Group:** The padding group works similar to the Margins Group when the node selected is a `< td>` or a cell in a table.
- **Background Group:** The Background group specifies the background image or color for a page.
    - **Background Image:** Click on the icon next to field to open file manager to select a background image.
    - **Background Color:** Color picker that lets the user define the background color to apply to the block element.



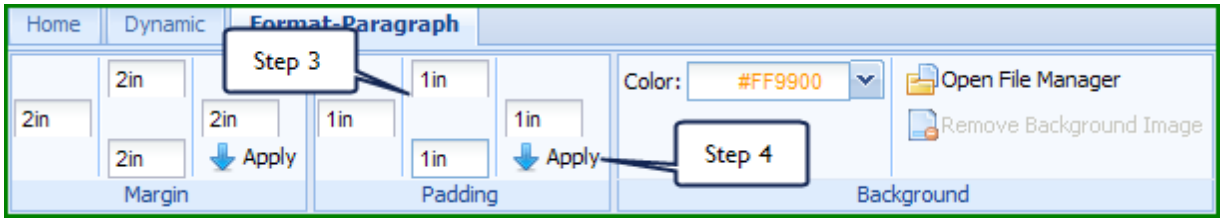
Background Images or Color is not supported when the output is of RTF Type.

Formatting Actions

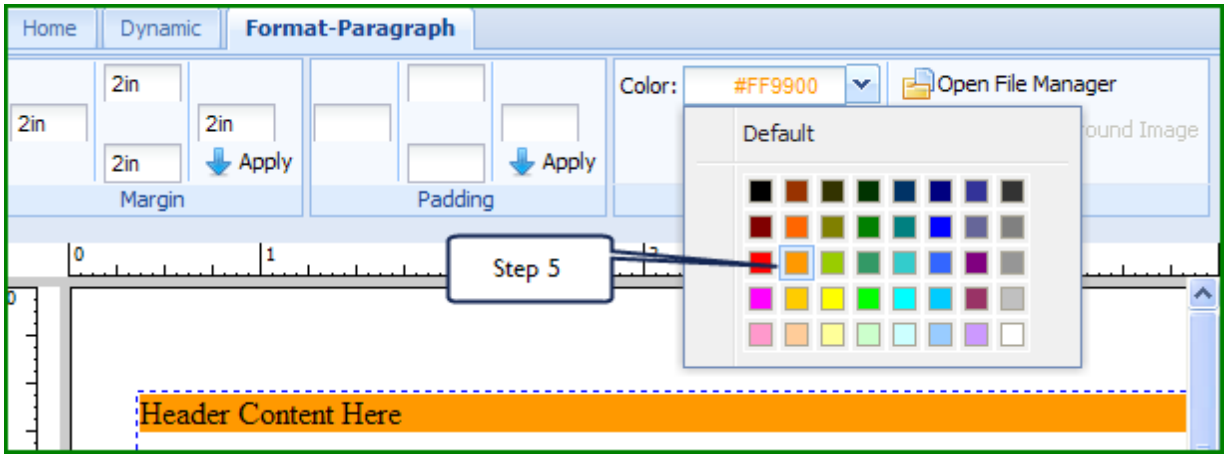
1. Select a field within the Margins group (top, bottom, left or right). Enter the size of the margins. *Each field should have a number, followed by the appropriate units. Valid units are: in, cm, mm. pt and px.*
2. Select **Apply**.



3. Select a field within the Padding group (top, bottom, left, or right). *This works similar to the Margins Group when the node selected is a `<td>` or a cell in a table.*
4. Select **Apply**.

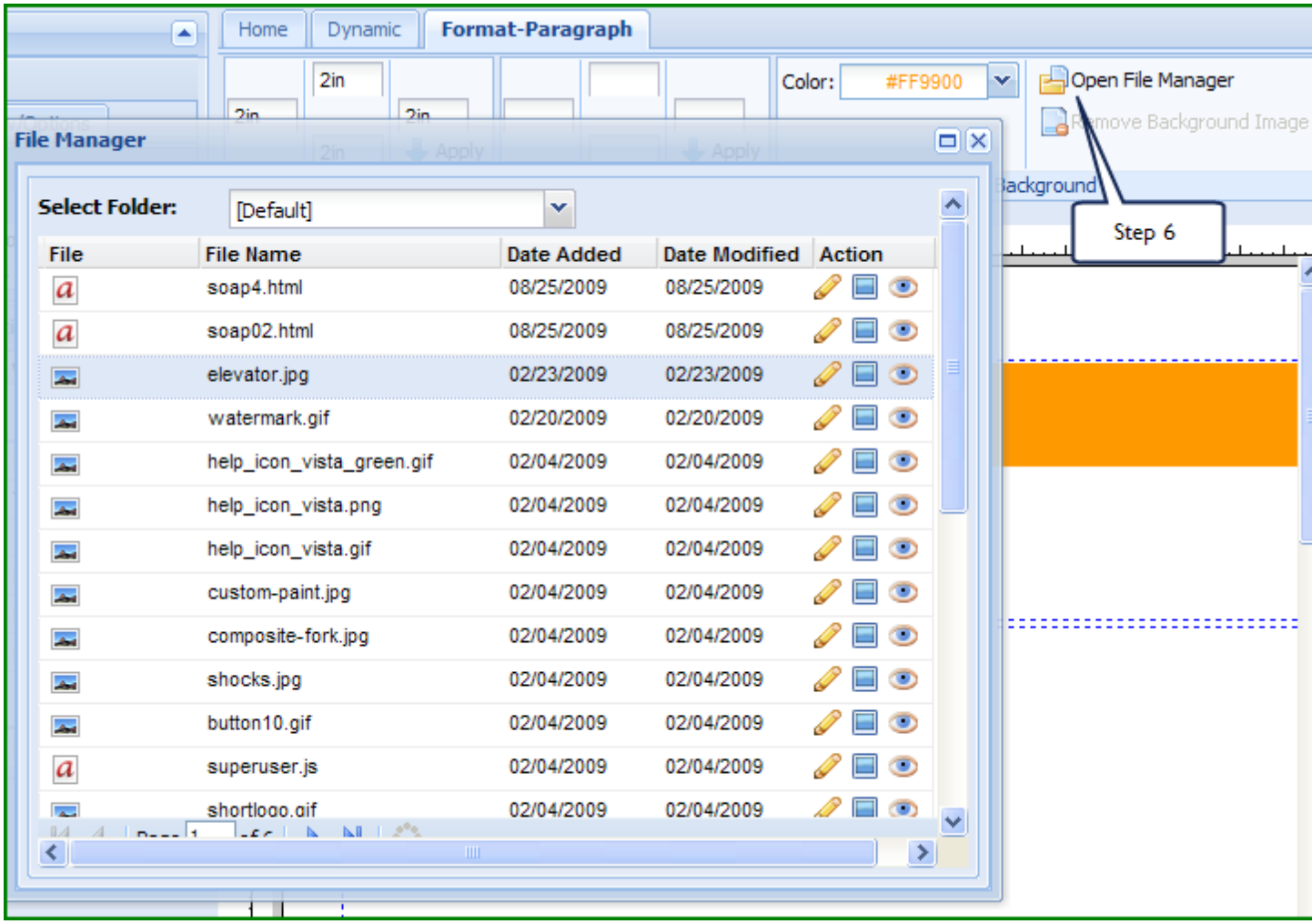


5. Select a background color. *This color will apply to the block element.*



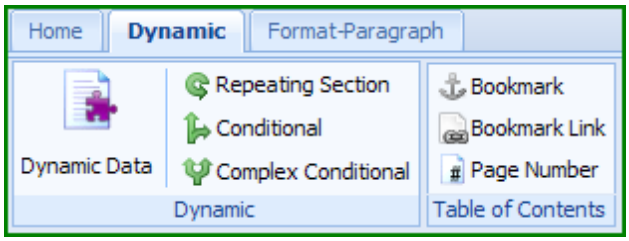
6. Select **Open File Manager** to select a background image. *This will open up the File Manager dialog box.*
    - *Select the pencil icon to insert the file link.*
    - *Select the photo icon to insert the file image.*

      Select the eye icon to preview the file image.



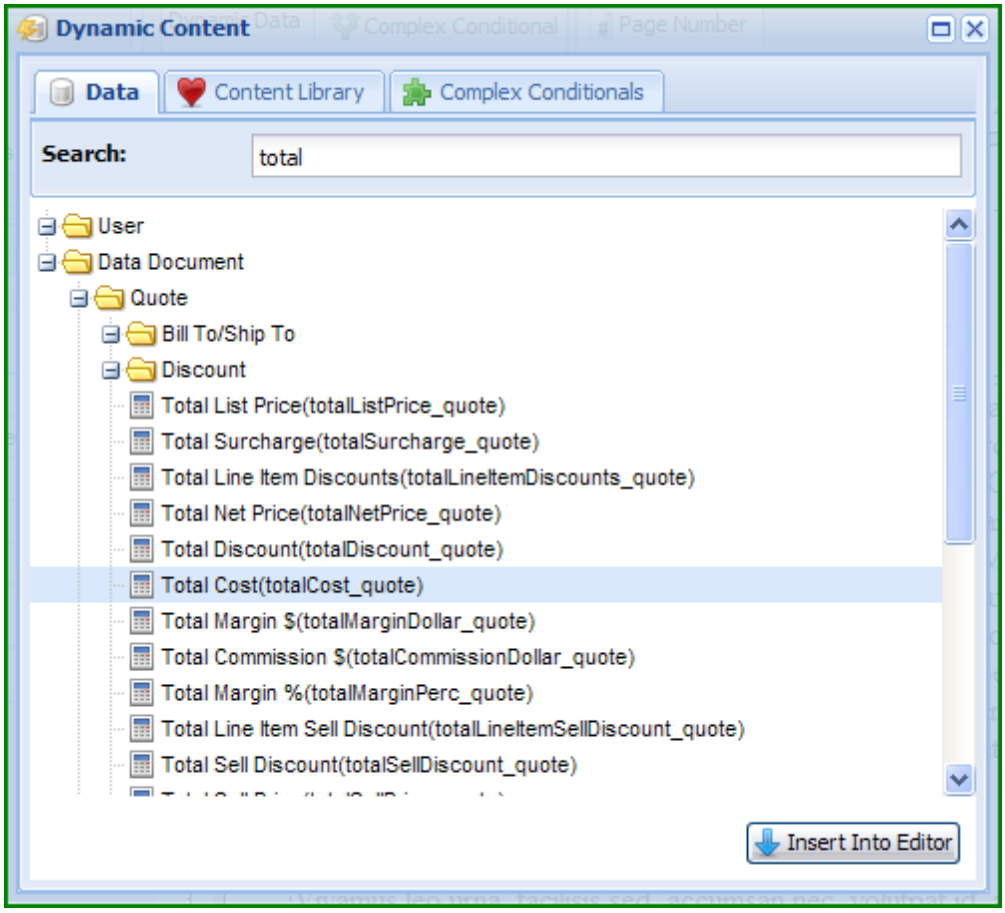
Dynamic Tab

You can access a host of tools that allow the user to customize page content. This group contains actions that allow the user to enter dynamic elements. The user can also manage loops, conditionals and complex conditionals. The Table of Contents Group helps manage bookmarks and page numbers.



Dynamic Group

This group contains actions that allow the user to enter dynamic elements. The user can also manage loops, conditionals and complex conditionals.

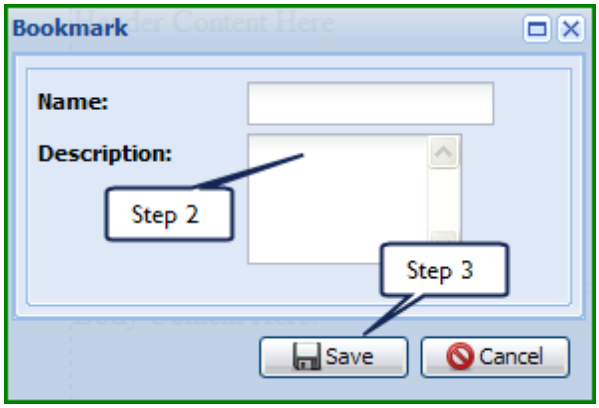


- **Dynamic Data:** Clicking on this action launches the Dynamic Content Window. This allows the user to access commerce attributes as well as the Content Library and the Complex Conditionals Library.
- **Repeating Section:** This action inserts a repeating loop that will be displayed repeatedly, depending on the conditions of the loop.. The content inside of loop can contain dynamic elements. This content can also be formatted. For more on adding/editing loops, see Loop Expression Editor.
- **Conditional:** You can insert a conditional if you want to define some content that will be displayed if a certain condition is met. The selected content is basically wrapped by an {if...then} statement. For more information on conditionals, see Conditional Expression Editor.
- **Complex Conditionals:** You can insert a complex conditional statements with multiple if...else clauses in-line within the content of a document without having to resort to custom XSL snippets. See Complex Conditional Library for more information.

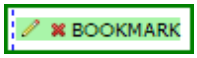
Table of Contents Group: Bookmarks

Bookmarks are internal links inside of your document. You can insert bookmarks anywhere in your content and create dynamic hyperlinks and page numbers to reference them. This allows you to create a basic Table of Contents. **To insert a bookmark:**

1. Select the **Bookmark** button. *This launches the Bookmark dialog.*
2. Enter a **Name** and **Description**. *The Name is used to refer to this bookmark from other places in the document and therefore need to be unique within a document. The description is only for admin purposes. This action inserts an anchor at the selected spot.*



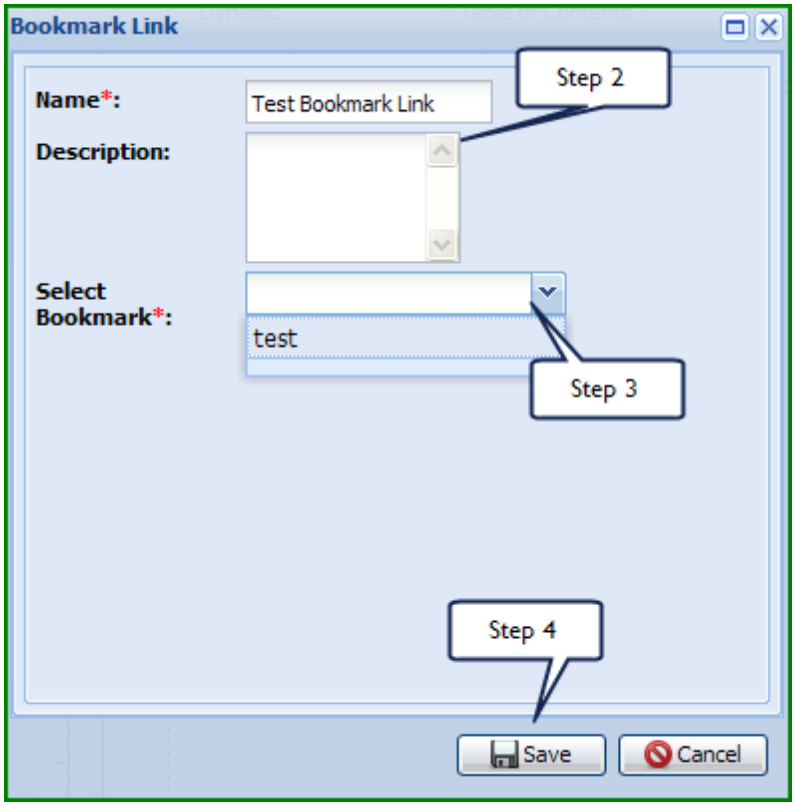
3. Select **Save**.
4. Bookmark elements are highlighted in green in the rich text editor. *Click the pencil to edit and the "x" to delete.*



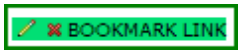
If you delete a bookmark, any other references to it will not function anymore.

Table of Contents Group: Bookmark Link

1. Select the **Bookmark Link** button.
2. Enter a **Name** and **Description**.
3. Choose the bookmark that the current selection will be linked to from a drop-down menu.



4. Select **Save**.
5. Double-click on the "X" to delete a Bookmark Link. *Bookmark Link elements are highlighted in dark green in the rich text editor.*



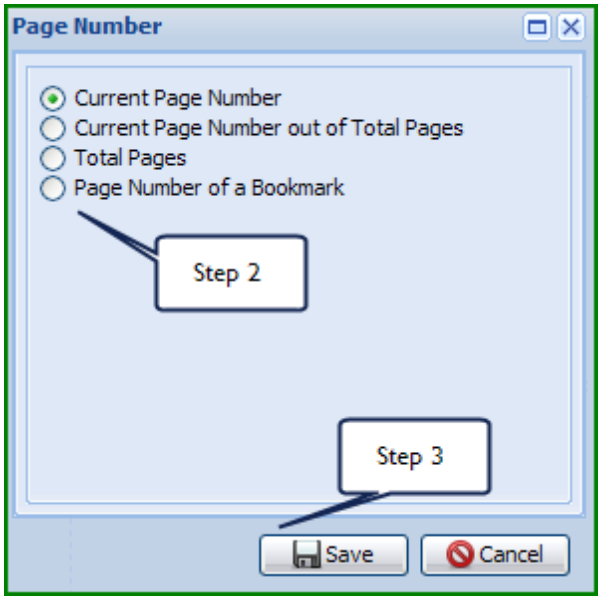
Click on the Bookmark Link button when you want to add some text that will link to a bookmark location when the user clicks on it. It inserts a hyperlink that references an existing bookmark.

You cannot edit a bookmark link.

Table of Contents Group: Page Number

Click on this action to enter a Page Number.

1. Place your cursor at a point in the Page editor, where you would like to insert the page number and select **Page Number**.
2. Select from four types of page numbers:
    - **Current Page Number:** Inserts a marker at the cursor point that displays the current page number on the output.
    - **Current Page Number out of Total Pages:** Displays the current page out of the total number of pages on the output.
    - **Total Pages:** Prints the total number of pages in the document.
    - **Page Number of Bookmark:** Prints the page number of the page that contains a specified bookmark (used for Table of Contents).
3. Select **Save**.



4. Double-click the "X" to delete Page Number element. *Page Number elements are highlight in light blue in the rich text editor.*

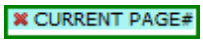
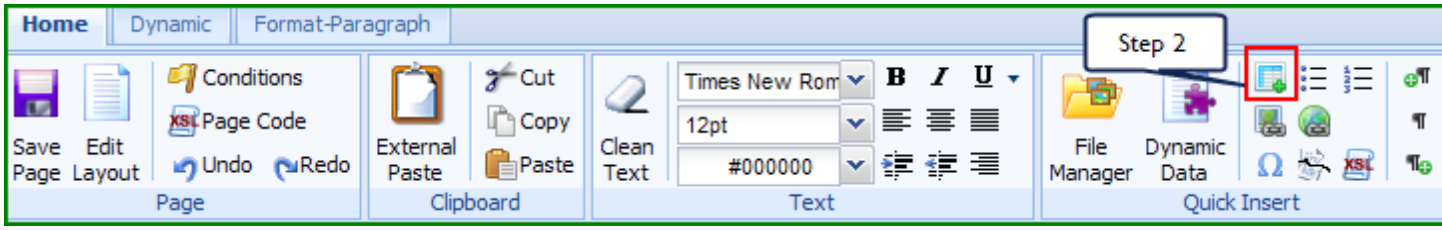


Table Tab

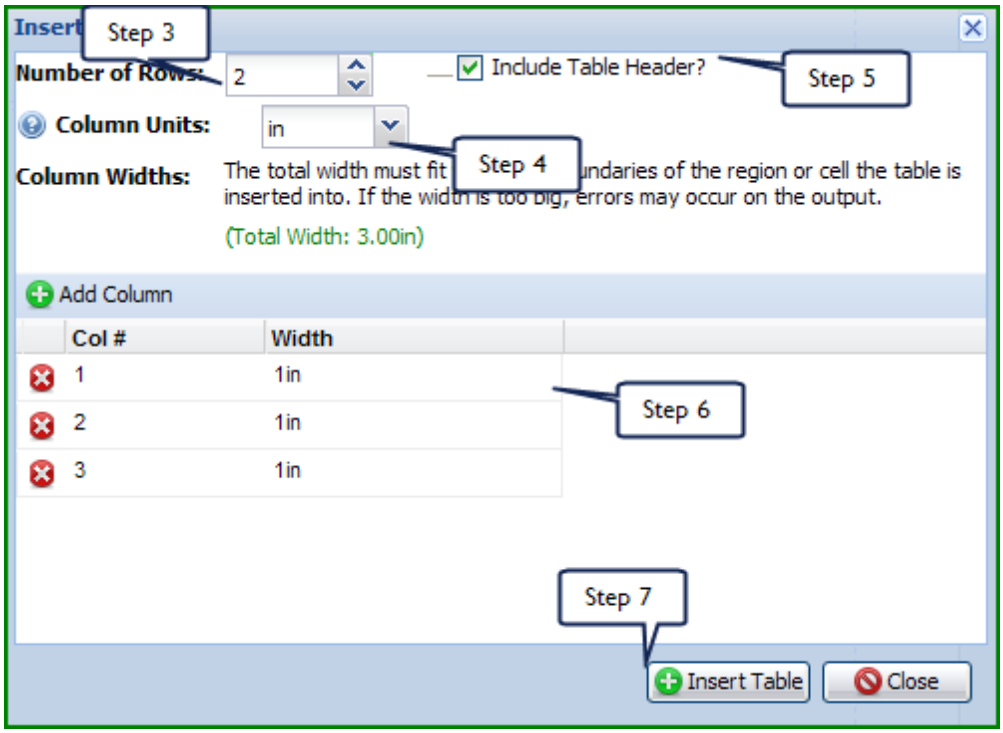
The Table tab contains actions that allow the user to edit Table Properties. This tab is visible only when the cursor is placed inside the table.Placing your cursor over an icon displays it's name.

Inserting a Table

1. Move your cursor to a location where you want to insert the new table.
2. Select the **Insert Table** icon the Home Toolbar. *The Insert Table dialog will appear.*



3. Determine the number of rows for the table.
4. Enter the number of columns you need for your table.
5. Select the box next to "Include Table Header" if you would like to include a header row in your table. *This header row will appear with a darker dotted border in the editor. If the contents of the table flow onto the next page, the table header row will repeat on the next page as well.*



6. Determine the width of each column (inches(in), centimeters(cm) or millimeters(mm)).
7. Select **Insert Table** or **Close** to close the dialog box without saving the changes or entering a table.

Editing a Table

After your table as been created, you will see the Table Tab appear:

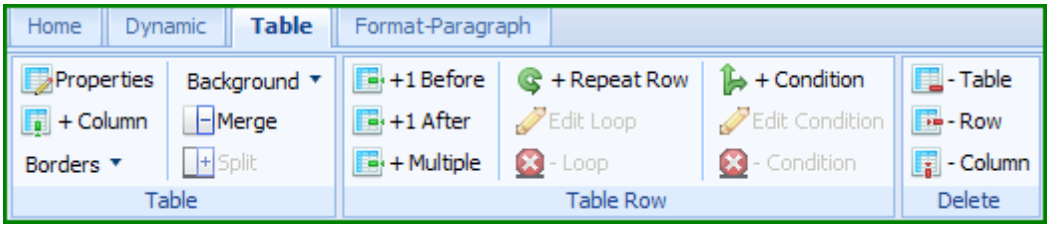
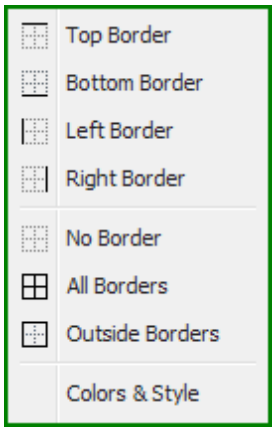
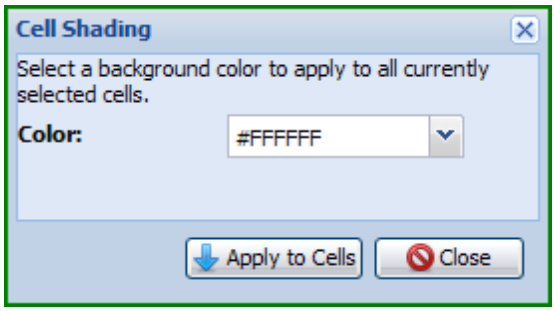


Table Groups

1. Select **Properties** to open the Edit Table Properties dialog. *Here you can make changes to columns units, the column width and you can choose to include or not include a table header row. What you can't change is the number of rows.*
2. Click the **+Column** icon to add a column to your table.
3. If you want to add borders to a cell, select **Borders**. From the Borders menu, select the type of border you want to apply. *You can also select the color and style of the border.*



4. Select the cells that you want to apply a background color and select **Background** from the menu. *Select "Set Cell Background" to open the Cell Shading dialog box.*
5. Select the desired color from the **Color** field.
6. Select **Apply to Cells**.



7. If you wish to merge cells, select the cells you want to merge together. *Hold down CTRL as you click the cells to merge and then click "Merge" in the toolbar. Repeat for Split.*

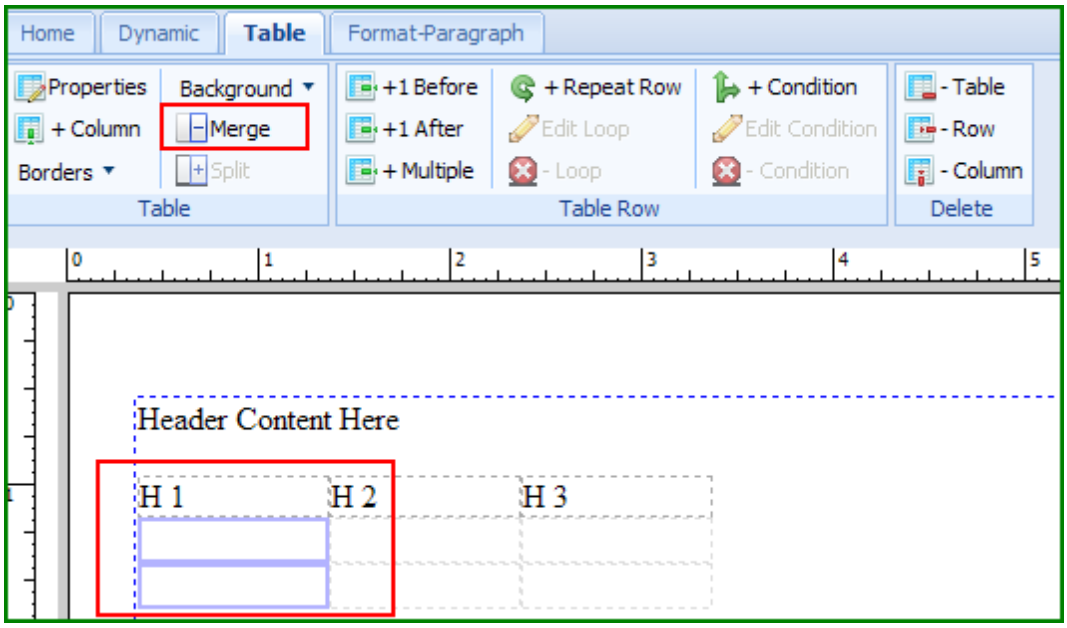
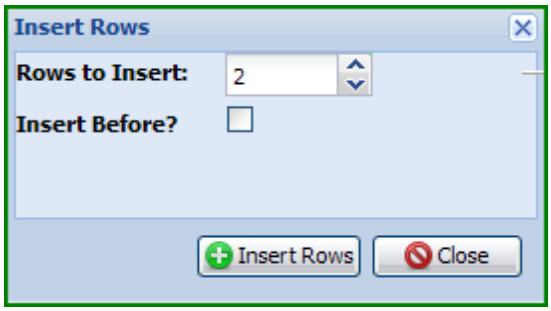


Table Row Groups

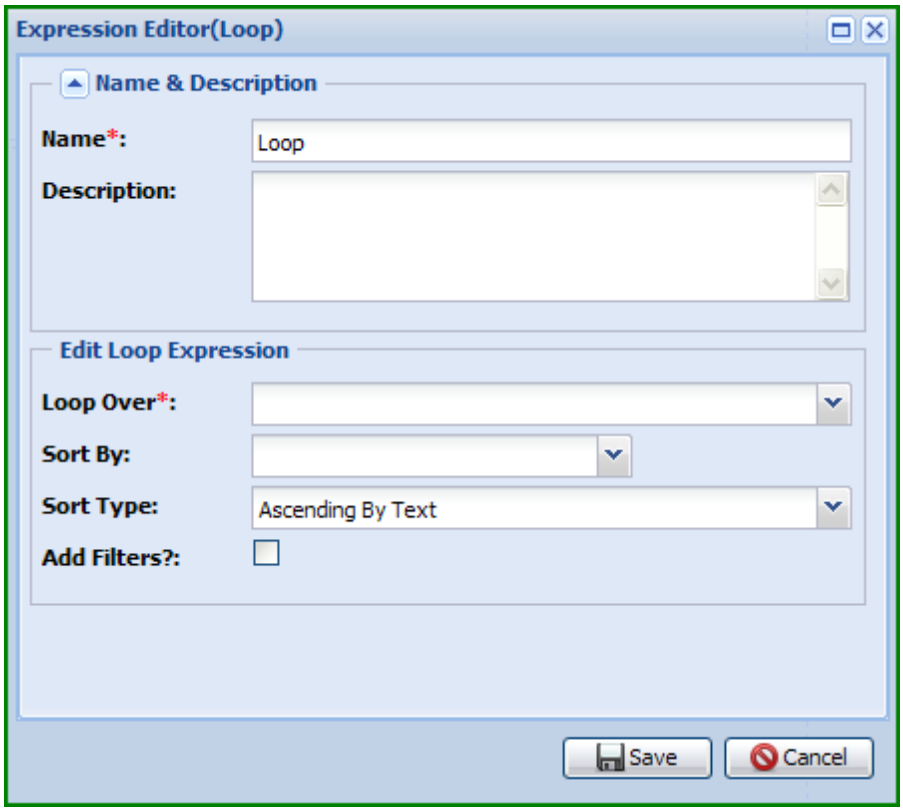
1. You can add rows to a table in three different ways:
   - **+Before:** Instantly adds a row before the cursor row.
   - **+After:** Instantly adds a row after the cursor row.
   - **+Multiple:** Opens the "Insert Rows" pop-up and allows you to enter:
     - **Rows to Insert:** The number of new rows you want to add.
     - **Insert Before:** Checked to insert the new rows BEFORE the cursor row. The default is to insert the new rows after the cursor row.



2. If you want a table row to be displayed for each line item in a transaction, place your cursor inside any table cell within the row that you want to repeat and select **+Repeat Row**. *The table row must not already be a repeating row or have an*

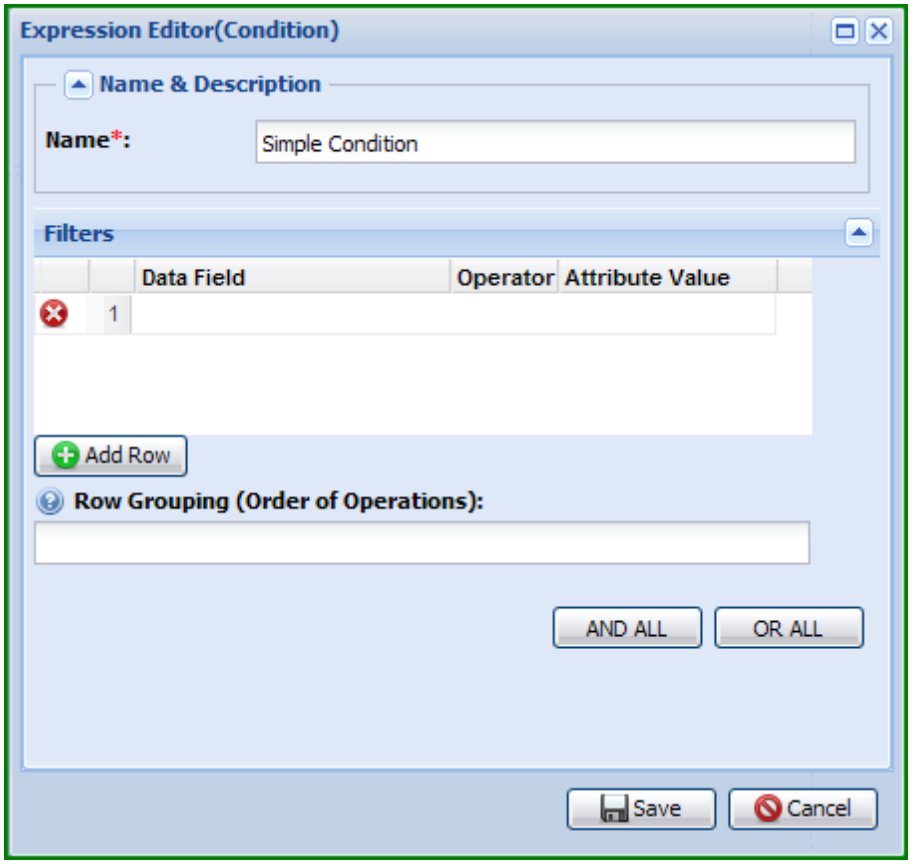
existing condition on it.

3. The **Loop Expression Editor** should appear syo ou can define the loop to display on the output.



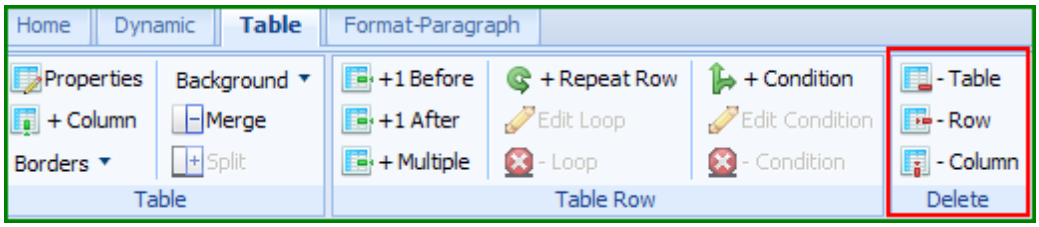
4. If you want to add a condition to a table row, place your cursor inside any table cell and select **+Condition**.

5. The **Condition Expression Editor** should appear and you can define conditions to be met for this table row to display on the output.



Deleting a Group

From the Delete group, you can delete a Table, Row or column.



To delete a row when the table editor is open, click on the "x" icon next to the column number that you want to remove.



Once a Loop or Condition has been added to the table, "Edit Loop", "Delete Loop", "Edit Condition" and "Delete Condition" will become visible on the Table Tab.

RELATED TOPICS

DOCUMENT ENGINE & XSL

OVERVIEW

Document Engine and XSL work hand-in-hand to allow you to create your desired document output.

XSL Code Window

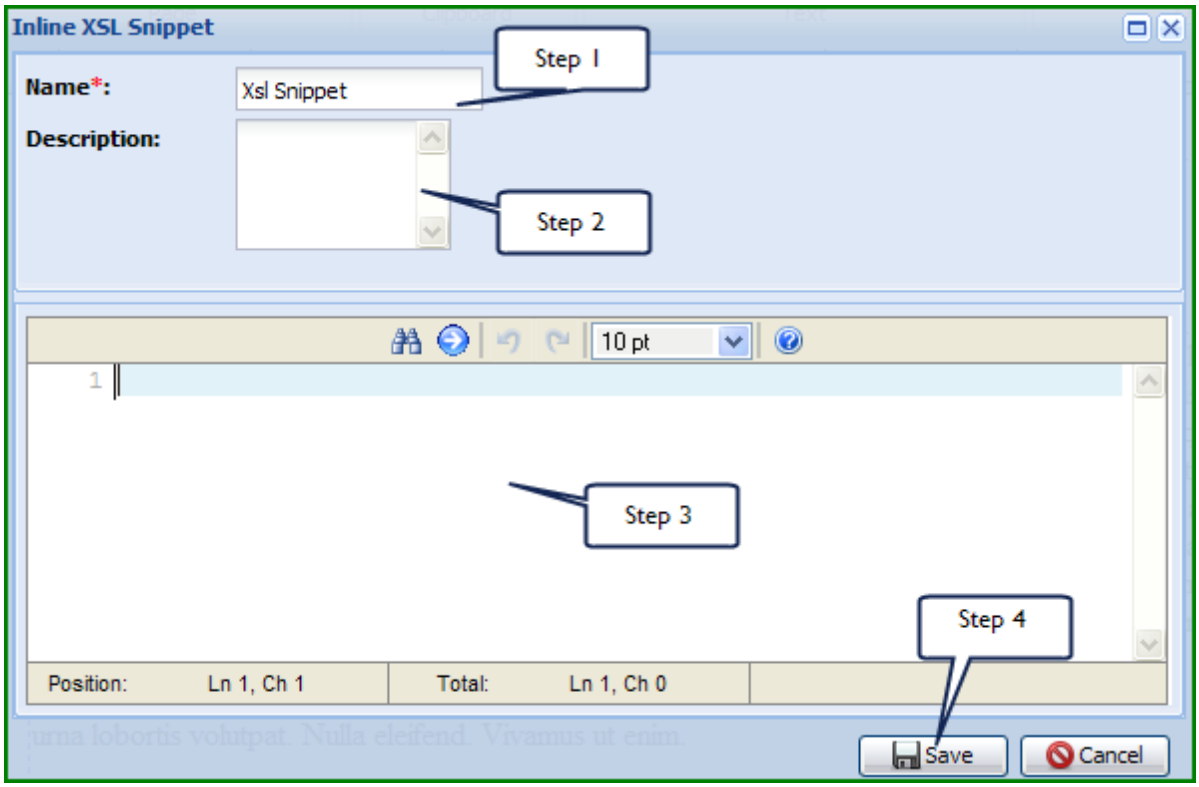
The XSL Code Window enables the users to edit custom XSL code within the BigMachines Document Engine. **The XSL Code Window can be invoked in the following ways:**

- **To add/edit Global XSL Snippet** - Select the Edit Global XSL Snippet (Document> Edit Global XSL Snippet) action on the Document Application Panel.
- **To add/edit Flow XSL Snippet** - Navigate to the Home Tab and select Insert XSL under Quick Insert.
- **To add/edit in line XSL Snippet** - Select Add XSL action on the Document Application Panel.

Using the XSL Code Window

1. **Name** the snippet.
2. Enter a **Description** for your XSL Snippet.
3. Write the custom XSL code in the editor. *Use the actions provided in the XSL Code Window tool bar to help in writing the code.*

4. **Save** the XSL Snippet.



Custom XSL

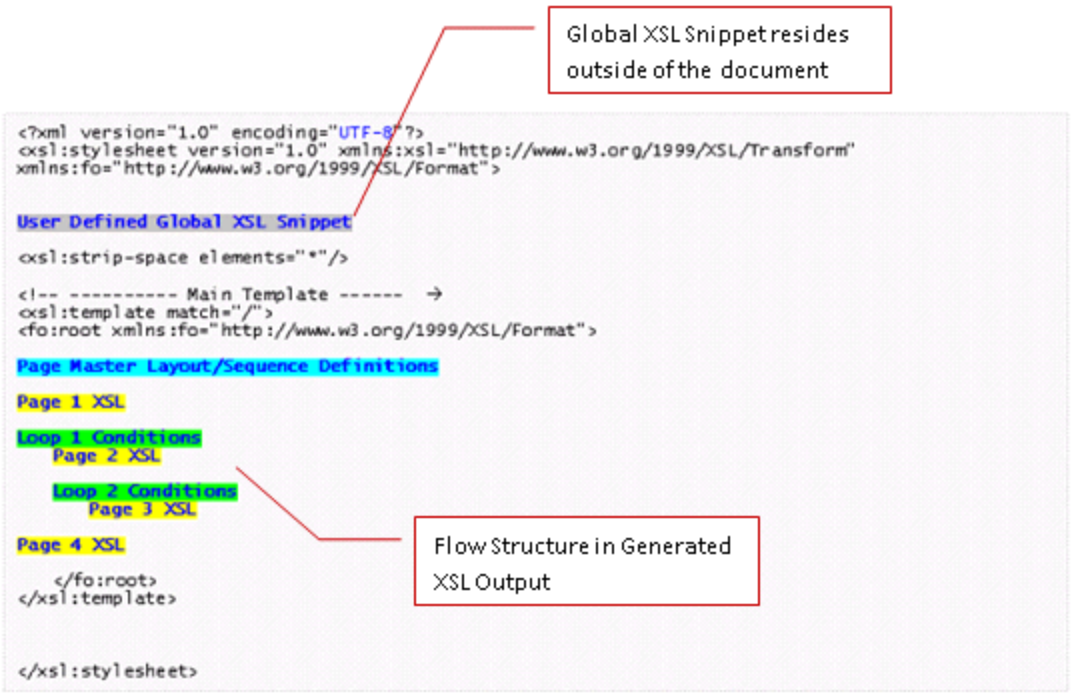
The BigMachines Document Editor (BMDE) reduces the need for writing custom XSL code to produce fully customized and branded commerce documents. If the users wish to further customize their documents, BMDE provides the opportunity to edit custom XSL code using the XSL Code Window. **Three types of custom XSL in a document:**

- **Global XSL Snippet** - This custom XSL snippet resides outside the scope of the main document.

> The Global XSL Snippet is empty by default. However, the users can add custom XSL code in this section by clicking the Edit Global XSL Snippet (Document > Edit Global XSL Snippet) action on the Document Panel.

- **InLine XSL Snippet** - The inline XSL Snippet is custom XSL that can be embedded within a page. To invoke the XSL Code Window and insert an inline XSL snippet in a page, navigate to the Home Tab and select the "Insert XSL" action under Quick Insert.

- **Flow XSL Snippet** - The Flow XSL can be inserted between the pages. Adding or editing this type of custom XSL is independent of editing a page in the Page Design Mode. To add this XSL, select "Add XSL" on the Document Panel.



Printer-Friendly XSL Views

The XSL view created on this page can be used to format print forms, email forms, and change history logs.

Adding Printer-Friendly XSL Views

1. Navigate from: Admin Home Page > Commerce and Documents > Process Definition > Printer-Friendly/History XSL Views
2. Enter a **Name**, **Variable Name** and **Description** for the XSL View Name.
3. Specify how you wish to process your output by choosing a value from the **Process Output Through** drop-down menu. The options are:
    - No Converter
    - No Converter - XML
    - FO to RTF Converter
    - FO to DOCX Converter
    - FO to PDF Converter

- FO to XLS Convertor. *The FO to XLS Convertor option is only available if the user chooses to upload a custom XSL.*

4. If you have selected the FO to PDF Convertor option, then you can either lock the generated PDF or keep it unlocked. To lock the PDF output, check the **Is PDF Read-Only** box.

5. You can either use the BigMachines Document Engine to edit the XSL file or upload a custom XSL.

   - **Editor:** Select this option if you are using the Document Engine.

   - **Upload XSL:** Select this option to upload a custom XSL File for the view. To do so, select **Browse** to use a file from your local system using the File Upload dialog. If you choose to upload a custom XSL, you must select the **Commerce XML** version. It is highly recommended that you choose the **New XML** version.

6. Choose the Document from the drop-down menu that will be associated with this printer-friendly view. *Only deployed documents will be available for selection.*

7. Select **Add** to save the view.



Adding a document and a printer-friendly XSL will NOT make your document available to the user. Remember to create a print action and link to the XSL View that you have created here.

Editing Printer-Friendly XSL Views

The XSL view modified on this page can be used to format print forms, email forms, and change history logs.

1. Modify the **XSL View Name** or the **Description**.

2. Choose a value from the **Process Output Through** drop-down menu. *The options are: No Converter, FO to RTF Converter, FO to PDF Converter or FO to XLS Convertor.*



The FO to XLS Convertor option is only available if Upload XSL is chosen in the Edit Mode section.

3. Edit Mode: You can either use the BigMachine Document Editor to edit the XSL File and then deploy it, or upload a Custom XSL.

4. Click **Display File** to have a preview of the XSL file.

5. Enter a **Transaction ID** to preview how the XSL file formats the actual data for **Document**, **History**, **Document XML**, and **History XML**.

6. Select **Apply** to apply the changes made and refresh the page. Select **Update** to update the changes made and go back to the page for managing printer friendly and history XSL views. Select **Back** to go back without saving.

DocX Documents & iPads

DocX documents are not available for use on iPad. It is possible to view RTF files if the file has been emailed to someone via an e-mail action. The steps below use the build-in mail application, Mail, which is standard on the iPad.

1. Go to the App Store and download the free app doctape.
2. Go to Mail and open the e-mail containing the document you wish to view.
3. Press and hold the attachment icon until a box pops up containing the options 'Quick Look' and 'Open in "doctape"'.
4. Select 'Open in "doctape"'. The app will upload the attachment then take you to a list of all documents opened through doctape. The one you just uploaded will be at the top.
5. Select the document you want to open.



In the Mail app, if you just click the attachment or select 'Quick Look' through the popup options, a new screen will open but it will just be a blank white page. *If this happens, you can simply click the action button in the upper-right corner (it looks like a box with a curved arrow pointing to the right) and then select 'Open in "doctape"'*



Disclaimer: This is what we tested with. We are not responsible for the functionality of any 3rd party application.

ADMINISTRATION

Restoring "Old" to XSL View

On the Printer Friendly/History XSLs, there is a bullet on existing XSLs that lets you choose between old and new XML version. If you accidentally select "New" and Apply, then you lose the ability to use the Old XML for that specific view. Fortunately, with FireBug you can get the option back:

1. Edit the HTML.
2. Input the option into the form.
3. Insert the following HTML into the form below the table row containing **Edit Mode**:

```
<tr class="bgcolor-form">
<td> </td>
<td>
<div id="preSuf" style="display: block;">
<table>
<tbody><tr class="bgcolor-form">
<td class="form-label" nowrap="true">
XML Version:
</td>
<td class="form-input" colspan="3">
Old
<input value="0" class="form-input" name="xml_version" checked="true" type="radio">
New
<input value="1" class="form-input" name="xml_version" type="radio">
</td>
</tr>
```

4. Click **Apply** to restore the functionality.

Setting the currency code dynamically

You can define an XSL variable when using a Commerce quote-level Attribute to set the currency code for all currency-formatted data in your Document.

1. Define the $main_doc variable in the Global XSL Snippet as shown in the Best Practices .
2. On the next line of the Global XSLSnippet, insert the following code:

```
<xsl:variable name="currencyCode"><xsl:value-of select="$main_doc/outputCurrency_quote"/></xsl:variable>
```



`outputCurrency_quote` is a commerce attribute holding the currency code.

3. Select **Currency** as the formatting within the Expression Editor.
4. Place `$currencyCode` directly into the currency-code field instead of entering the actual currency code.



This is standard in the Base Template. *Be sure to familiarize yourself with what is included by default in the Global XSL Snippet.*

"currencyValue" template

If applying currency formatting in Global XSL or in snippets, calls can be great simplified by wrapping a thin layer around the `BMI_universalFormatPrice` template. The following template's (`currencyValue`) only parameter is formatting the number. It applies the same conversion rate, currency code, and other formatting options every time; the parameters do not need to be passed/set every time a currency is formatted.

```
<xsl:template name="currencyValue">
<!-- takes number parameter and performs rounding, adds appropriate currency separators and symbols -->
<xsl:param name="value" />
<xsl:variable name="numValue" select="number($value)" /> <!--validate input is a number, and fix sci notation -->
<xsl:variable name="decimalNum">
<xsl:call-template name="convertSciToNumString">
<xsl:with-param name="inputVal" select="$numValue"/> <!-- supposedly not necessary with number(), but done for safety-->
</xsl:call-template>
</xsl:variable> <!--xsl:value-of select="round($decimalNum*100.0) div 100.0"/-->
<xsl:choose>
<xsl:when test="$numValue = 'NaN'"><xsl:value-of select="$value"/></xsl:when> <!-- echo back the value arg if it's not a number -->
<xsl:otherwise>
<xsl:call-template name="BMI_universalFormatPrice">
<xsl:with-param name="price" select="$decimalNum" />
<xsl:with-param name="currency" select="$currencyCode" />
<xsl:with-param name="multiplier" select="$conversionRate" />
<xsl:with-param name="showCurrencySymbol">true</xsl:with-param>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

The following template incorporates a call to `convertSciToNumString`. *Adding this template to the Global XSL snippet ensures that scientific notation will not affect documents adversely.*

```
<xsl:template name="convertSciToNumString">
<xsl:param name="inputVal" select="0"/>
<xsl:variable name="numInput">
<xsl:value-of select="translate(string($inputVal),'e','E')"/>
```

```
</xsl:variable>
<xsl:variable name="maxExp">
<xsl:value-of select-
t="'0000000000
```

```
<xsl:text>1</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:variable>
<xsl:value-of select="number($numCoefficient) * number($numMantisa)"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

Best Practices

- Ensure that these are the first two lines of your Global XSL Snippet:

```
<xsl:variable name="main_doc" select="/transaction/data_xml/document[@document_var_name='quote_process']/" />
<xsl:variable name="sub_doc" select="/transaction/data_xml" />
```



This is standard in the Base Template. *Be sure to familiarize yourself with what is included by default in the Global XSL Snippet.*

- Add a Flow XSL Snippet when looping over Model Line Items using the Page Loop in the flow. *Use the following code:*

```
<xsl:variable name="model_line" select="."/>
```



This allows you to always have an XPath (`$model_line`) with which to access configurable attributes for the current Model, even if the current model is out of scope on the page(s) in the loop.

USE CASES

Using a multi-select menu to attach PDFs to an output Document

Using a recursive template, it is possible to attach multiple PDFs to an output document, based on the user's selections in a multi-select menu. Using the following method, which is site agnostic and optimized for a multi-language output, only one multiselect menu is required. In the example below, the value of the multi-select menu is used to construct the file's URL in the File Manager. *The file format is [langaugeAbbreviation][MSM Variable Value].pdf.*

1. Add the following code to the **Edit Custom Master Layouts** page in the Document Engine:

```
<fo:simple-page-master master-name="pdfPage" page-height="11in" page-width="8.5in" margin-top="0in" margin-bot-
tom="0in" margin-left="0in" margin-right="0in">
<fo:region-body/>
</fo:simple-page-master>
<!-- Page sequence -->
<fo:page-sequence-master master-name="pdfPageSequence">
<fo:repeatable-page-master-alternatives>
<fo:conditional-page-master-reference master-reference="pdfPage" page-position="any"/>
</fo:repeatable-page-master-alternatives>
</fo:page-sequence-master>
```

2. Add the following templates to the Global XSL Snippet:

```
<xsl:template name="printablePDF">
<!-- outputs a page with a background image from the URL passed
if URL is blank, this will output a blank page. This is used to attach data sheets. PDFs will output the full number
of pages
These are generally stored in the file manager. Use Edit Custom Master Templates in doc engine to modify master-ref
pdfPageSequence-->
<xsl:param name="insertURL"/>
<fo:page-sequence content-width="95%" content-height="95%" background-image="{$insertURL}" background-repeat-
t="paginate" master-reference="pdfPageSequence">
<fo:flow flow-name="xsl-region-body">
<fo:block><!--xsl:value-of select="$insertURL" /--></fo:block>
</fo:flow>
</fo:page-sequence>
</xsl:template>
```

```
<xsl:template name="genDataSheets">
<!-- this template breaks aprart includeDataSheet_quote attribute and builds a file URL from each value.
NOTICE: menu value must match filename of PDF MINUS language prefix and file extension, which are appended here-->
<xsl:param name="sheetList" />
<xsl:param name="sheetLanguage" />
<xsl:variable name="delim">~</xsl:variable> <!--delimiter between values in includeDataSheet_quote, which is a multi-
select menu-->
<xsl:variable name="fileExt">.pdf</xsl:variable> <!-- fileExt could be made into a parameter for outputting image
pages, etc-->
<xsl:variable name="thisDataSheet">
<xsl:choose>
```

```
<xsl:when test="contains($sheetList, $delim)"><xsl:value-of select="normalize-space(substring-before($sheetList,$de-
lim))" /></xsl:when>

<xsl:otherwise><xsl:value-of select="$sheetList" /></xsl:otherwise>

</xsl:choose>

</xsl:variable>

<xsl:if test="$thisDataSheet !=''">

<!-- NOTE: beware of file name/location or language changes - check file manager or go to url built here if data sheet
is missing-->

<xsl:call-template name="printablePDF">

<xsl:with-param name="insertURL" select="concat('/bmfsweb/', $main_doc/site_quote, '/image/Proposal/', 'en',
$thisDataSheet, $fileExt)"/>

<!-- !!!MD TODO: english-only data sheets until translated ones come in - then replace 'en' above with $sheetLanguage.
-->

</xsl:call-template>

</xsl:if>

<xsl:variable name="rest" select="substring-after($sheetList,$delim)" />

<xsl:if test="$rest != ''">

<xsl:call-template name="genDataSheets">

<xsl:with-param name="sheetList" select="$rest" />

<xsl:with-param name="sheetLanguage" select="$sheetLanguage" />

</xsl:call-template>

</xsl:if> <!--xsl:if test="$rest = ''"></xsl:if-->

</xsl:template>
```

3. Create a custom XSL page in the Document Engine that calls the recursive template, passing the value of the multi-select menu, as well as the language with the following code. *`includeDataSheet_quote` is the variable name of the multiselect menu in Commerce.*

```
<!-- This snippet will attach any data sheets selected (assuming the files reside in file manager -->

<!-- NOTE: see globalXSL snippet for template definition. See also Custom Master Layout used by the template -->

<xsl:if test="$main_doc/includeDataSheet_quote != ''">

<xsl:call-template name="genDataSheets">

<xsl:with-param name="sheetList" select="$main_doc/includeDataSheet_quote" />

<xsl:with-param name="sheetLanguage" select="$main_doc/outputLanguage_quote" />

</xsl:call-template>

</xsl:if>
```



The value of the multi-select menus in Commerce is a tilde-delimited list of the selected values.

This can only be done with PDF files on PDF output documents. *Use Commerce Attribute, `site_quote`, to specify the site-specific portion of the file's URL within the File Manager.*

Attaching images in the document using the file attachment Attribute

The following example XSL Snippet is used to attach an image in the document. This image is dynamically uploaded to the Quote by the user utilizing the file attachment Attribute.

```
<xsl:variable name="main_doc" select="/transaction/data_xml/document[@document_var_name='quote_process']"/>

<xsl:if test="string-length($main_doc/logo_quote) > 0">

<xsl:variable name="attachLogo" select="concat('http://', $main_doc/creatorCompany_quote, '.big-
machines.com/servlet/ImageServer?&amp;file_type=cmAttachmentAttr&amp;file_id=',$main_doc/logo_quote)"/>

<fo:block>

<fo:external-graphic src="{$attachLogo}" content-height="1.5in" height="1.5in"/>

</fo:block>

</xsl:if>
```

`creatorCompany_quote` is a Commerce attribute which has the value of lower Attribute, `_system_supplier_company_name`. This is used to have relative path in the Document Engine. *`logo_quote` is the file attachment attribute used.*

Implementing toUpper & toLower

You may want to make a String all uppercase or lowercase before performing comparisons or String operations. The following is a simple way to convert a String to all lowercase using `toLower`

```
<xsl:param name="strToConvert" />

<xsl:value-of select="normalize-space(translate($strToCon-
vert,'ABCDEFGHIJKLMNOPQRSTUVWXYZÀÁÂÃÄÅÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜCC', 'abcde-
fghijklmnopqrstuvwxyzàáâãäåèéêëìíîïñòóôõöùúûücc'))" />

</xsl:template>
```

If the second and third parameters in "translate" are reversed, then this template becomes `toUpper` and the String converts to all uppercase:

```
<xsl:template name="toUpper">

<xsl:param name="strToConvert" />

<xsl:value-of select="normalize-space(translate($strToConvert, 'abcde-
fghijklmnopqrstuvwxyzàáâãäåèéêëìíîïñòóôõöùúûücc', 'ABCDEFGHIJKLMNOPQRSTUVWXYZÀÁÂÃÄÅÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜCC'))" />

</xsl:template>
```



This works with most accented Roman characters.

NOTES



BigMachines supports generating document and proposal outputs as Microsoft Word, or DocX files. Users will have the ability to generate DocX output using document engine, custom XSL and print action in Commerce.



Free apps that were tested and found to be incompatible: OliveOfficeBasicHD, FileApp, PlainText, SimpleNote.

RELATED TOPICS

USE CASE - DYNAMICALLY ADDING SIGNATURES IN THE DOCUMENT ENGINE

OVERVIEW

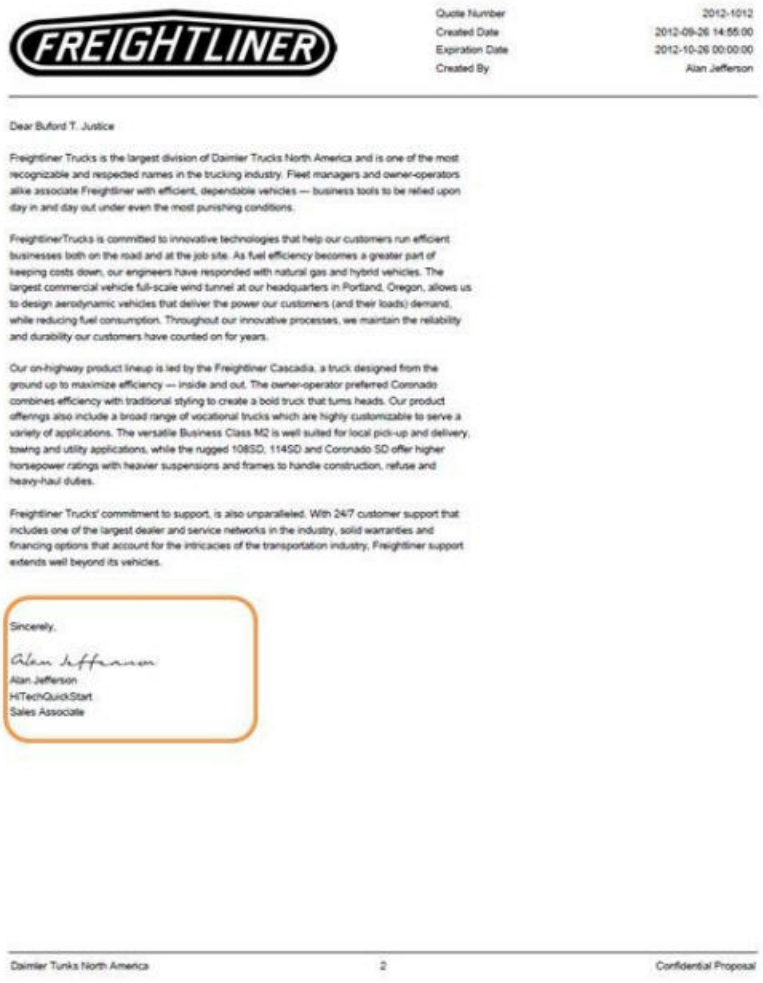
You will need to use the file manager, commerce attributes and the Document Engine to dynamically add signatures to each document. Before beginning this tutorial, you should understand how to use the File Manager, BML and the Document Engine.

Prerequisites

Before using this tutorial, you must:

- Know how to create commerce attributes
- Understand XSL and BML
- Be a Blue Belt Level Administrator

See Sample Document



FREIGHTLINER

Quote Number	2012-1012
Created Date	2012-09-26 14:55:00
Expiration Date	2012-10-26 00:00:00
Created By	Alan Jefferson

Dear Buford T. Justice

Freightliner Trucks is the largest division of Daimler Trucks North America and is one of the most recognizable and respected names in the trucking industry. Fleet managers and owner-operators alike associate Freightliner with efficient, dependable vehicles — business tools to be relied upon day in and day out under even the most punishing conditions.

FreightlinerTrucks is committed to innovative technologies that help our customers run efficient businesses both on the road and at the job site. As fuel efficiency becomes a greater part of keeping costs down, our engineers have responded with natural gas and hybrid vehicles. The largest commercial vehicle full-scale wind tunnel at our headquarters in Portland, Oregon, allows us to design aerodynamic vehicles that deliver the power our customers (and their loads) demand, while reducing fuel consumption. Throughout our innovative processes, we maintain the reliability and durability our customers have counted on for years.

Our on-highway product lineup is led by the Freightliner Cascadia, a truck designed from the ground up to maximize efficiency — inside and out. The owner-operator preferred Coronado combines efficiency with traditional styling to create a bold truck that turns heads. Our product offerings also include a broad range of vocational trucks which are highly customizable to serve a variety of applications. The versatile Business Class M2 is well suited for local pick-up and delivery, towing and utility applications, while the rugged 108SD, 114SD and Coronado SD offer higher horsepower ratings with heavier suspensions and frames to handle construction, refuse and heavy-haul duties.

Freightliner Trucks' commitment to support, is also unparalleled. With 24/7 customer support that includes one of the largest dealer and service networks in the industry, solid warranties and financing options that account for the intricacies of the transportation industry, Freightliner support extends well beyond its vehicles.

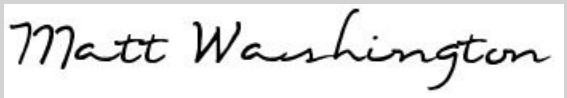
Sincerely,

Alan Jefferson
HiTechQuickStart
Sales Associate

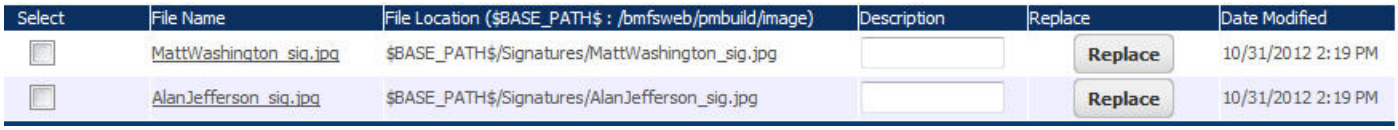
ADMINISTRATION

Step 1: Uploading Signatures to the File Manager

1. Create a folder in the File Manager to store your signature images. In this case, the folder will be called "**Signatures**".
2. Create a **signature image (.jpg)** for each user that will have a signature included on some document.
3. Name the file after the users' name without spaces.

	User's Name: Alan Jefferson Filename: AlanJefferson.jpg
	User's Name: Matt Washington Filename: MattWashington.jpg

4. Upload the image files into the 'Signatures' folder in the File Manager.



Select	File Name	File Location ($BASE_PATH$: /bmfsweb/pmbuild/image)	Description	Replace	Date Modified
☐	MattWashington_sig.jpg	$BASE_PATH$/Signatures/MattWashington_sig.jpg		Replace	10/31/2012 2:19 PM
☐	AlanJefferson_sig.jpg	$BASE_PATH$/Signatures/AlanJefferson_sig.jpg		Replace	10/31/2012 2:19 PM

Step 2: Creating Commerce Attributes

1. Navigate to the Documents List of the commerce process you will use.
2. Create the following **main-document commerce attributes:**
   - **Signature Exists** (signatureExists_quote); Boolean Attribute
   - **SignatureURL** (signatureURL_quote); Text
3. Create a commerce library, **Dynamic Signatures**, for the commerce process you will use.
4. Select input attributes:
   - **System Attribute**: _system_supplier_company_name
   - **Main Document Attribute**: preparedByName_quote
5. Enter the following BML script. Click here to see sample script.

```
returnString = "";

signatureExists = false;
```

```
//Set the URL for the signature file
file = replace(preparedByName_quote, " ", "") + ".jpg";
signatureURL = "http://"
+ _system_supplier_company_name
+ ".bigmachines.com/bmfsweb/"
+ _system_supplier_company_name
+ "/image/signatures/"
+ file;

//Script determine whether the signature exists or not
error = "NO FILE";
response = urldatabypost(url, "", error);
signatureExists = response <> error;

//Return string
returnString = "1~signatureExists_quote~" + string(signatureExists) + "|" + "1~signatureURL_quote~" + sig-
natureURL + "|";
```

Step 3: Updating Print Actions

1. Run the Library Function on the Print Actions Add the following to the Advanced Modification function of these actions:
   - **View Proposal PDF**(generateProposal)
   - **View Proposal RTF**(generateProposalRTF)
2. This script should be added right before the semicolon in the return statement:

```
+ commerce.dynamicSignatures();
```

Step 4: Modifying Document Engine Document

1. Modify the Global XSL Snippet of the Document by adding the following line to the top of the Global XSL Snippet in the Document Engine if it does not already exist:

```
<xsl:variable name="main_doc" select="/transaction/data_xml/document[@document_var_name='quote_process']/"/>
```

2. Add the Signature Image to the Document
3. Choose the location of the signature in the Document Engine
4. In this location, create a Conditional Section with the condition: Signature Exists = true
5. Insert an XSL snippet as the conditional content with the following XSL code:

```
<xsl:variable name="signatureImage" select="$main_doc/signatureURL_quote"/>
<fo:external-graphic src="{$signatureImage}" content-height=".2in" height=".2in"/>
```

RELATED TOPICS

EXPRESSION EDITOR

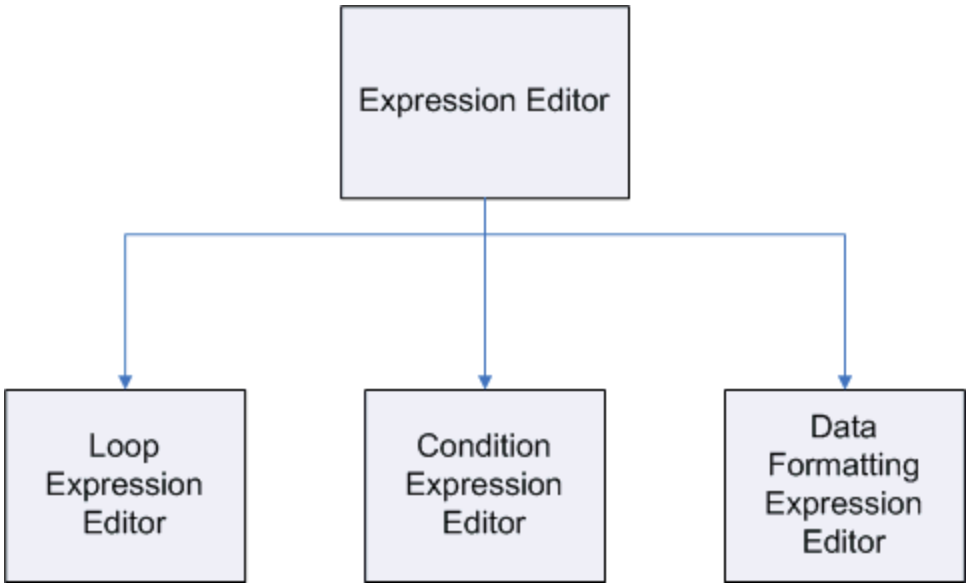
OVERVIEW

The Expression Editor is used to edit XSL expressions within a document. It allows the user to manipulate page elements using XSL functionality without having to resort to full-fledged coding.

Using the Expression Editor

Each variation of the expression editor is customized to support the specific document component.

- Common Properties:
  - **Expression Text Name** - Each Expression must have a text name.
  - **Expression Description** - The users can enter some description in this text area.
- Common Actions:
  - **Code View** - View the code that is generated by the expression. The users can directly modify the code. If code is modified, the users are prompted that saving code changes will convert this expression into custom XSL.
  - **Update** - The expression is validated and saved when this action is performed.



There are three variations of the Expression Editor:

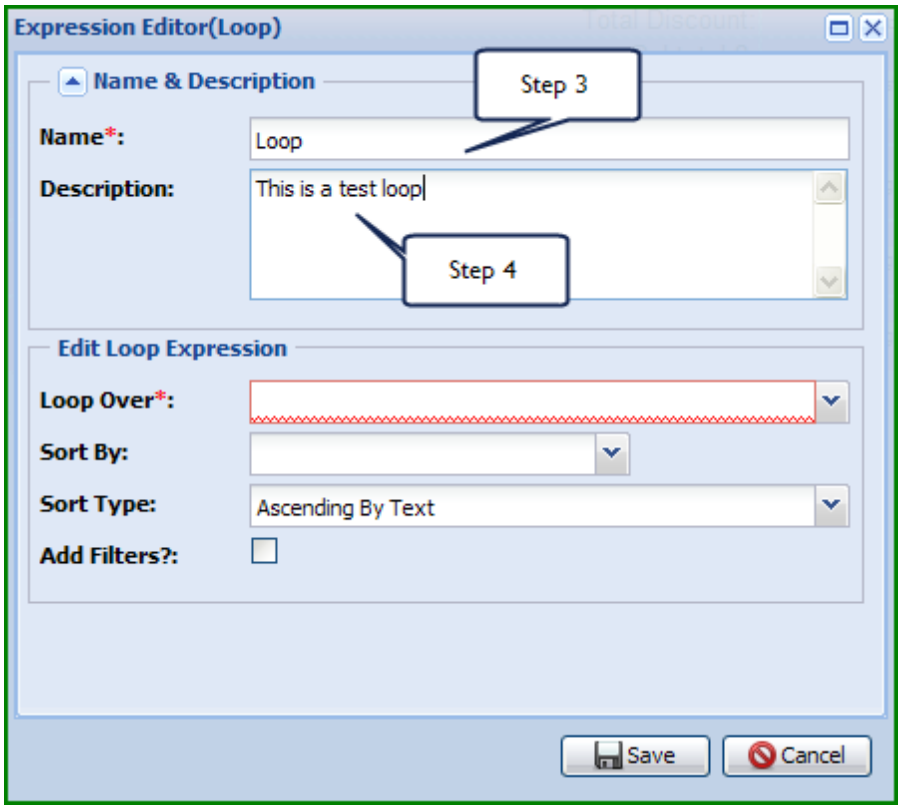
- Loop Expression Editor

Navigation: Document Editor > Dynamic Tab > Repeating Section

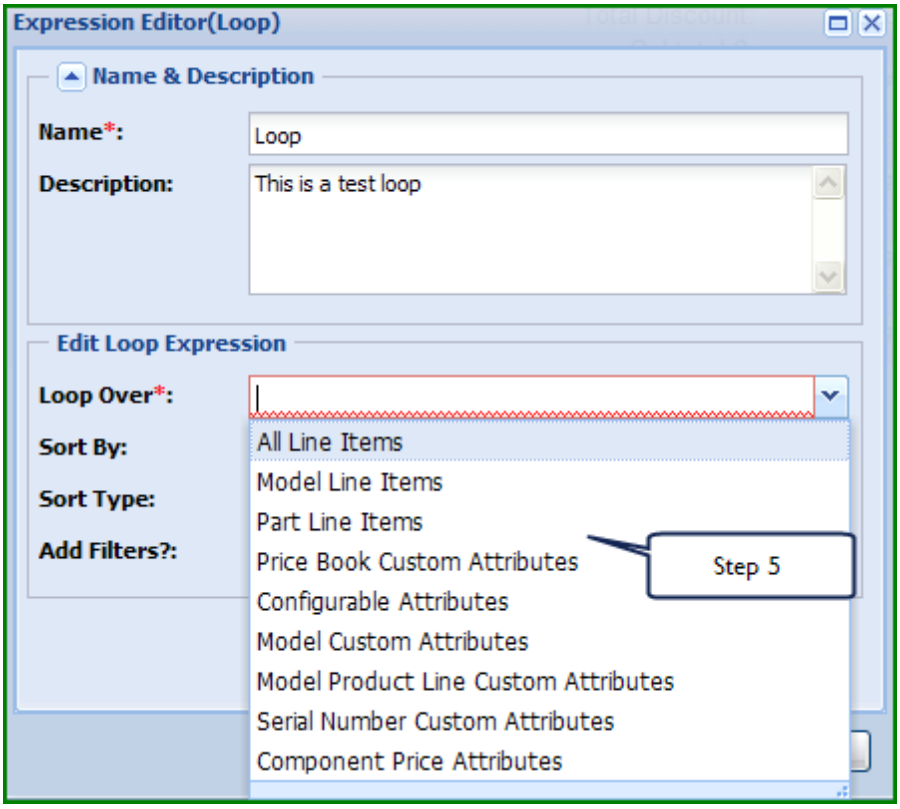
The Loop Expression Editor is used to define and edit a loop without having to resort to any coding. Loops can be edited within a page or a document.

Inserting a Loop

1. Place your cursor at a point in the Page Editor where you would like to insert the repeating section.
2. From the Dynamic Tab, select **Repeating Section** to open the Loop Expression Editor.
3. Enter a **Name** for the section.
4. Enter a **Description** for your repeating section.



5. Using the drop-down list, choose the data that is to be looped over from the drop-down menu. *Line items:*
    - **Model Line Items:** Only configured line items
    - **Part Line Items:** Only non-configured line items
    - **Price Book Custom Attributes:** Loop through the price book custom attributes (fields 1-20)
    - **Configurable Attributes:** All configurable attributes and values
    - **Model Custom Attributes:** All custom model attributes
    - **Model Product Line Custom Attributes:** All product line custom attributes
    - **Serial Number Custom Attributes:** All serial number custom attributes
    - **Component Price Attributes**



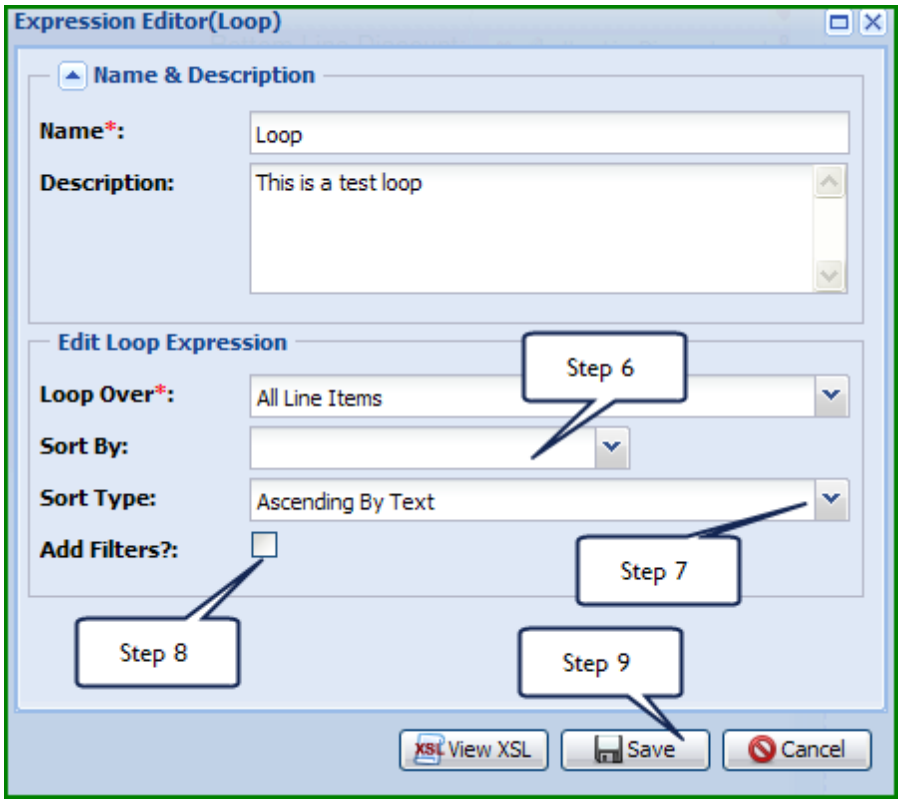
6. Using the drop-down list, choose which attribute should be used to sort the results. *The attribute that you select for **Sort By** needs to be a part of what data you are looping over.*

7. Select the method that will be used to sort based on the selected **Sort By** value. *The users can sort text or numbers in ascending order (A to Z, or zero to 9), or in descending order (Z to A, or 9 to zero).*

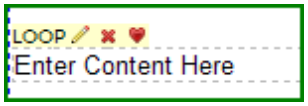
8. Select **Add Filters?** to specify the conditions that will filter out the data that the loop iterates over:

    - **Data Field:** Existing data columns that are search-able in commerce.
    - **Operator:** Enable the comparison of the data field with the value to search for and generate the filter result.
    - **Attribute Value:** Specify the value of the data field.
    - **Order of Operations:** Defines the full expression logic and logical operations between each condition attribute. Each row can be referenced by its index number assigned by the system. By default, all condition rows are AND-ed.
        - *Use And All button to AND all of the condition attributes in the order of their creation (index number).*
        - *Use the Or All button to OR all of the condition attributes in the order of their creating (index number).*
        - *You do not need to use any order for your attribute indexed while defining the order of operation. The attribute indexes can be referenced multiple times in the order of operation to define the correct condition. The order of operation defined is validated by the system when you click Update or View Code.*

9. Click **Save** to save and update the loop condition. *You will see the Loop section appear in your document, highlighted in light yellow.*



10. Insert the content to display where it reads **Enter Content Here**. *You can also insert more dynamic element (conditionals, dynamic data, loops, etc.) inside of the content.*



Click Add Row to add another row to the filter.

Editing a Loop

- **To edit a loop:** click on the Pencil icon and the Expression Editor will open. *You will see a button appear for "View XSL". Click to view the element's XSL.*
- **To delete a loop:** click on the red X.
- **To add a loop:** click on the "heart" icon.

Example Use Case: Inserting a Loop

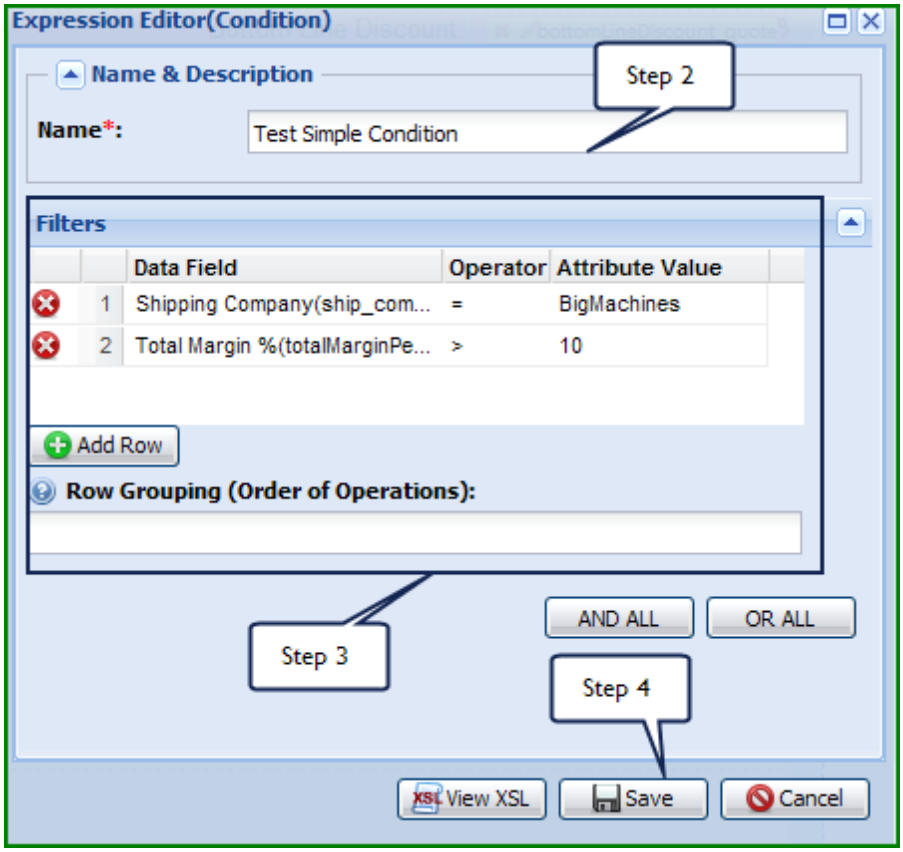
1. Output a list that contains the **Part Number** of all Part Type Line Items with Lead Time less than 5 days OR Part Base Price less than $100.
2. Specify What Data you want and how it will be displayed (build the table).
3. Create a list. *For the list element, specify a dynamic place holder for Part Number since this is the field you want to display in your list.*
4. Populate the fields you want with the right data from the Transaction (build the loop).
5. Select the list that you created above and launch the Loop Expression Editor by clicking on the **Repeating Section** action in the Dynamic Data tab.
6. Follow the steps listed above for "Inserting a Loop"
    - **For Loop Over:** Choose Part Line Items. This creates a loop that iterates through all the part type line items in your transaction.
    - **For Sort By:** Choose Part Number. This sorts the results in alphabetical order.
    - Create a filter that returns part line items with lead times less than 5 days or part base price less than $100:
        - *Choose the first data field as Lead Time. Specify that Lead Team < 5.*
        - *Choose the second data field as the User Currency Preference. Specify that Currency Preference = USD.*
        - *Choose the third data field as Part Base Price. Specify that Part Base Price < 100.*
        - *In the Row Grouping Field, group the conditions like this: (1 OR (2 AND 3)).*

- **Conditional Expression Editor**

This editor helps in specifying and editing conditions for the selected area in the page or document. If the condition is met, then the selected node(s) will be hidden in the output.

Defining a Conditional

1. Place your cursor at a point in the page where you would like to insert a conditional.
2. Enter a **Name** and **Description** for the condition.

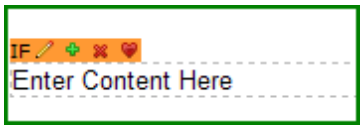


3. Add filters to specify the conditions. *Complete the following information:*

    - **Data Field:** Select from the existing data columns that are search-able in commerce.
    - **Operator:** Select from the list of operators in the field by clicking directly on the field. The operators enable the comparison of the data field with the value to search for and generate the filter result.
    - **Attribute Value:** Specify the value of the data field.
    - **Order of Operations:** Use this section to define the full expression logic and logical operations between each condition attribute. Each row can be referenced by its index number assigned by the system. By default, all condition rows are AND-ed.
        - *Use And All button to AND all of the condition attributes in the order of their creation (index number).*
        - *Use the Or All button to OR all of the condition attributes in the order of their creating (index number).*
        - *You do not need to use any order for your attribute indexed while defining the order of operation. The attribute indexes can be referenced multiple times in the order of operation to define the correct condition. The order of operation defined is validated by the system when you click Update or View Code.*

4. Click **Save** to save changes. *If this is a new condition, it will now appear on your document highlighted in Orange. Click Cancel to exit the Expression Editor without saving changes.*

5. In the IF element, insert the contents where it says, "Enter Contents Here". *You can also insert more dynamic elements (conditionals, dynamic data, repeating section, etc.) inside of the content.*



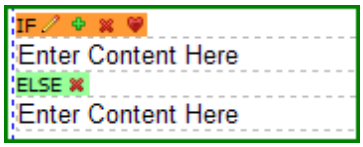
Click Add Row to add another row to the filter.

Adding an ELSE Clause

1. In the conditional "IF" section icon, double-click on the GREEN "+" button. *An ELSE block should appear below the "IF" content.*



2. Edit the content in the dotted box which will display if the condition in the "IF" is not met.



If you need to remove the ELSE block, simply click the "X" icon next to the ELSE label. *This will remove all content inside of the ELSE block.*

Example: Conditional Formatting

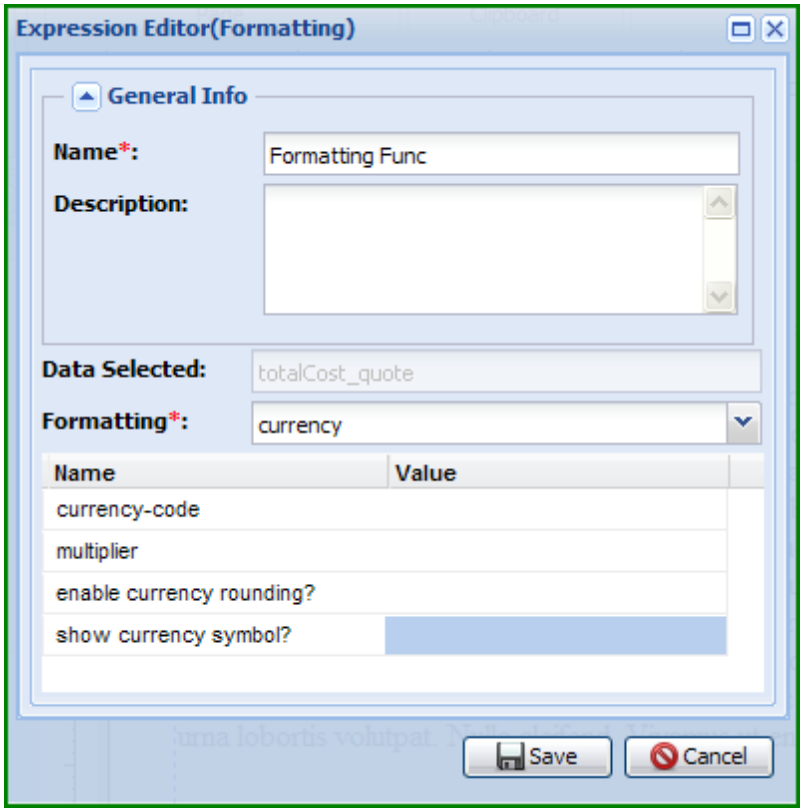
Show a part in the contract named "For California Buyers" to buyers from California only. Buyers from other states should not see this section in the contract.

1. Name the section: "For California Buyers" - use the rich text editor to author the content.
2. Hide the section: Select the "For California Buyers" text and launch the Conditional Expression Editor. In the Data Fields, choose the CRM Attribute "Buying State Province".
3. Specify the condition "Buying State Province = California".

When a buyer, who is not from the state of California logs in, the condition is met and the section is hidden. For a buyer from California, the condition is false and the text is therefore no longer hidden.

- **Data Formatting Expression Editor**

The Data Formatting Expression Editor is used to define and edit the format for dynamic data without having to resort to any coding.



Defining a Format

- **Source:** This indicates that the dynamic commerce data will be derived from the current process.
- **Name:** Enter a name for the dynamic data that you want to format. This is a required field. The text name is not unique.
- **Description:** Enter a valid description for the data to be formatted.
- **Data Selected:** This is a read-only field that indicates that the dynamic data field was selected.
- **Formatting:** Use the tree drop-down list to select the function(type) to be applied to the data. This function will be applied to the dynamic data from the commerce XML. The result of applying the function on the data is displayed on the output.

- **Update:** Click Update to update the loop condition in the database.
- **Cancel:** Click Cancel to cancel the loop condition definition.
- **Parameter Grid:** This section displays parameters of the formatting function. The number of parameters and the type of parameter depends on the function that is chosen in the Formatting section.
    - ***Name:*** *Provides a name for the parameter that is expected by the function picked in the Formatting field*
    - ***Value:*** *Edit the values of the parameters in the Value field.*

Currency Formatting

We support dynamic values for currency-code and value multiplier in printed documents. There are additional parameters that control rounding and display of currency symbol. Multi-currency applications can take advantage of this feature to print price and discount attributes in any supported currency regardless of the currency of the current user of the currency in which the document was initially created.

Currency Function Parameter	Description
Currency_code	The fullaccessuser can now pass a global XSL variable name in this field so that currency is shown depending on the logic specified by the fullaccessuser. The default value for this field is the current user's preferred currency.
Multiplier	The fullaccessuser can now pass a global XSL variable name in this field so that the correct multiplier is applied on the logic specified by the fullaccessuser in the global XSL snippet. The default value for this field is one.
Enable Currency Rounding	When enabled, this option will round the currency value. Be default, this option is set to false.
Show Currency Symbol	When enabled, this option will show the currency symbol depending on the currency specified in the currency-code parameter. Be default, this option is set to true.

Example: Data Formatting

Goal: The cover letter must be addressed to the Name of the Buyer. The name of the buyer is the user name contained in the email address. For example, if the user's email is dvader@empire.com, then the letter must be addressed to dvader.

Solution:

- *Specify what data you want:* Drag and Drop the **User Email Dynamic Data** at the selected node in the document page.
- *Format your data:*
    - In the **Formatting** section, choose substring-before.
    - In the **Value** section, specify "@". This returns the string before the "@". In our example, when the email is dvader-@empire.com, it returns dvader.

ADMINISTRATION

Formatting Expression Editor Parameters: When defining a format in the Formatting Expression Editor, use the tree drop-down list to select the function(type) to be applied to the data. This function will be applied to the dynamic data from the commerce XML. The result of applying the function on the data is displayed on the output.

General Functions

Function	Parameters	Description
normal-size	None	Trims any extra whitespace before and after the data value.
count	None	Returns the number of times the data value occurs.
sum	None	Returns the sum of the numerical value of each node in the data value.

String Functions

Function	Parameters	Description
substring	**start-index:** {number} starting position to take the substring from; **length:** {number} length of string to take	The substring function returns the substring of the data value starting at the start-index with length of length. For example, if the data value is "12345", then substring(2,3) returns "234".
substring-after	search string: {text}: starting position to take the substring from	The substring-after function returns the substring of the data value that follows the first occurrence of the search string, or the empty string if the search string was not found. For example, if the data value is "1999/04/01", the substring after("/") returns 04/01 and substring after("19") returns 99/04/01.
substring-before	search string: {text}: starting position to take the substring from	The substring-before function returns the substring of the data value that precedes the first occurrence of the search string, or the empty string if the search string was not found. For example, if the data value is "1999/04/01", then substring before("/") returns 1999.
concat	param 1: {text} string to concatenate to end of a data value	The concat function returns the concatenation of its arguments.
string-length	None	The string-length function returns the number of characters in the data value.
translate	**search-characters:** {text} starting position to take the substring from; **replace-characters:** {text} length of string to take	The translate function returns the data value with occurrences of characters in the search-characters parameter replaced by the character in the replace-character parameters. **For example:** if the data value is "bar", then translate("abc", "ABC") returns the string "BAr". If there is a character in the search-characters parameter with no character at a corresponding position in the replace-characters parameter, then occurrences of that character in the first argument string are removed. For example, translate("abc-", "ABC"), where the data value "---aaa--", returns "AAA". *If a character occurs more than once in the search-characters parameter, then the first occurrence determines the replacement character. If the replace-characters parameter is longer than the search-characters parameter, then excess characters are ignored.*

Numeric Functions

Most of these functions are standard XSLT functions. Detailed documentation on parameters can be found on XSLT reference web sites such as: http://www.devguru.com/Technologies/xslt/quickref/xslt_index_functions

Function	Parameters	Description
format-number	**format-number:** {text} *Required*- Specifies the format pattern; **decimal-format:** {text} *Optional*	The format-number function is used to convert the data value into a formatted string. Characters used in the formatting pattern: **#** - Denotes a digit; **0** - Denotes leading and following zeroes; **.** - Denotes a decimal point; **,** - The group separator for thousands; **%** - Displays the number as a percentage; **;** - Pattern separator. The first pattern will be used for positive numbers and the second for negative numbers.
currency	currency-code: {text} *Required*	Formats a number into a currency string.
round	None	Rounds the data to the nearest integer.
floor	None	Returns the largest integer that is not greater than the data value.
ceiling	None	Returns the smallest integer that is greater than the data value.

Date Functions

There are no predefined functions available to format dates as **dd.MM.yyyy** or **dd/MM/yyyy**. The fullaccessuser will have to write a custom XSL to achieve this formatting.

Function	Parameters	Description
shortdate	separator: {text} *Required*. The delimiter to use, "/" in the example to the right.	Formats a BigMachines date attribute into a short-hand readable format such as 2/12/2008.
longdate	separator: {text} *Required though not used.* Enter anything into this field.	Formats a BigMachines date attribute into a long-hand readable format, such as February 12, 2008.

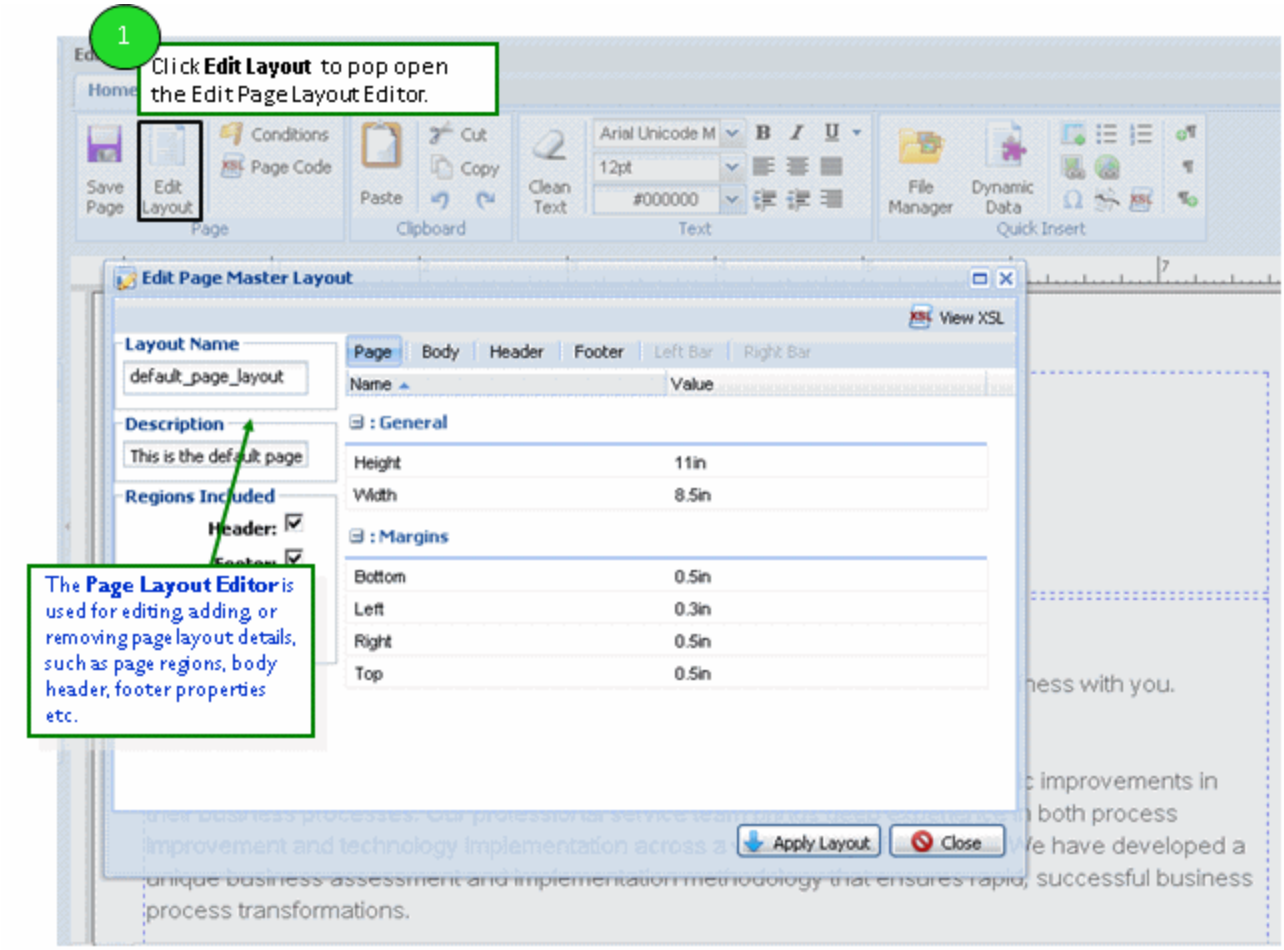
RELATED TOPICS

PAGE LAYOUT EDITOR

OVERVIEW

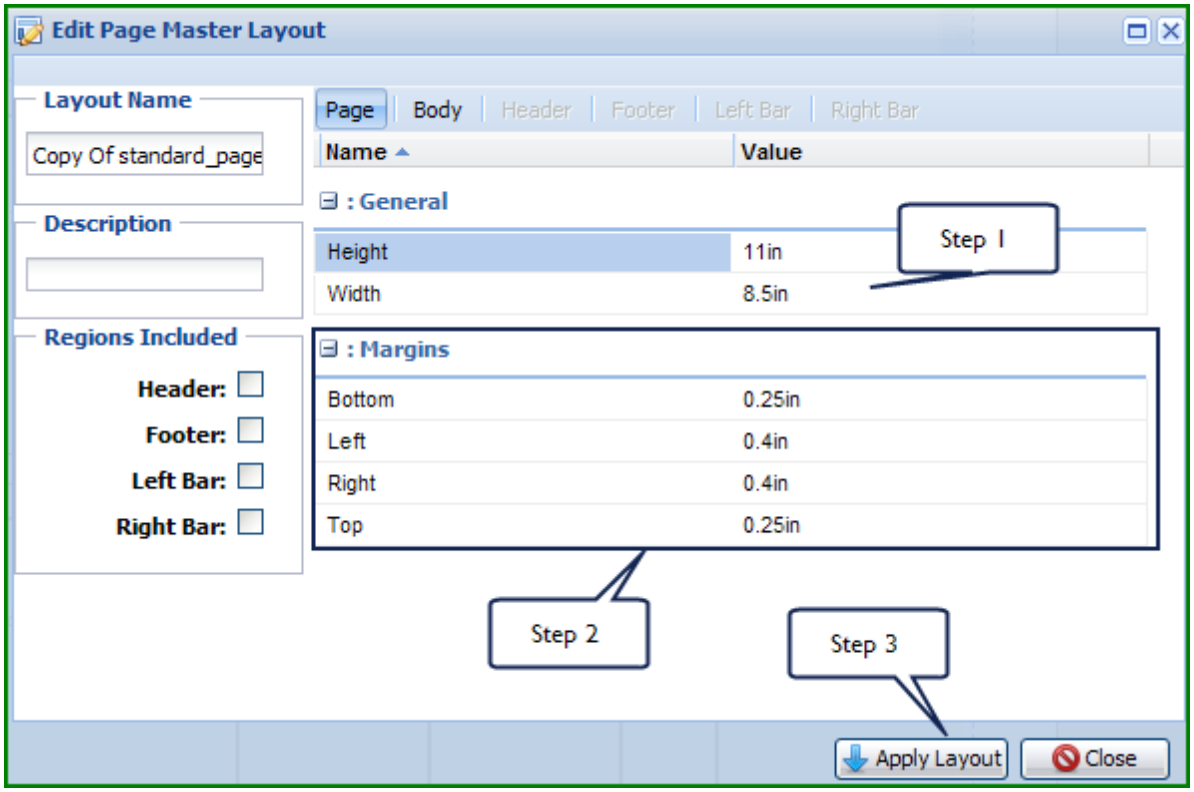
The Page Layout Editor is used for editing, adding, or removing page layout details, such as page regions, body header, footer properties etc.

Navigation: Document Editor > Home Tab > click Edit Layout button on Toolbar



Page Tab

1. Enter a value change to the height and width of the page. *The height and width set the page size.*
2. Use the four margin settings available in this section to change the default margins. *Do not forget to enter the unit of measure as a suffix.*

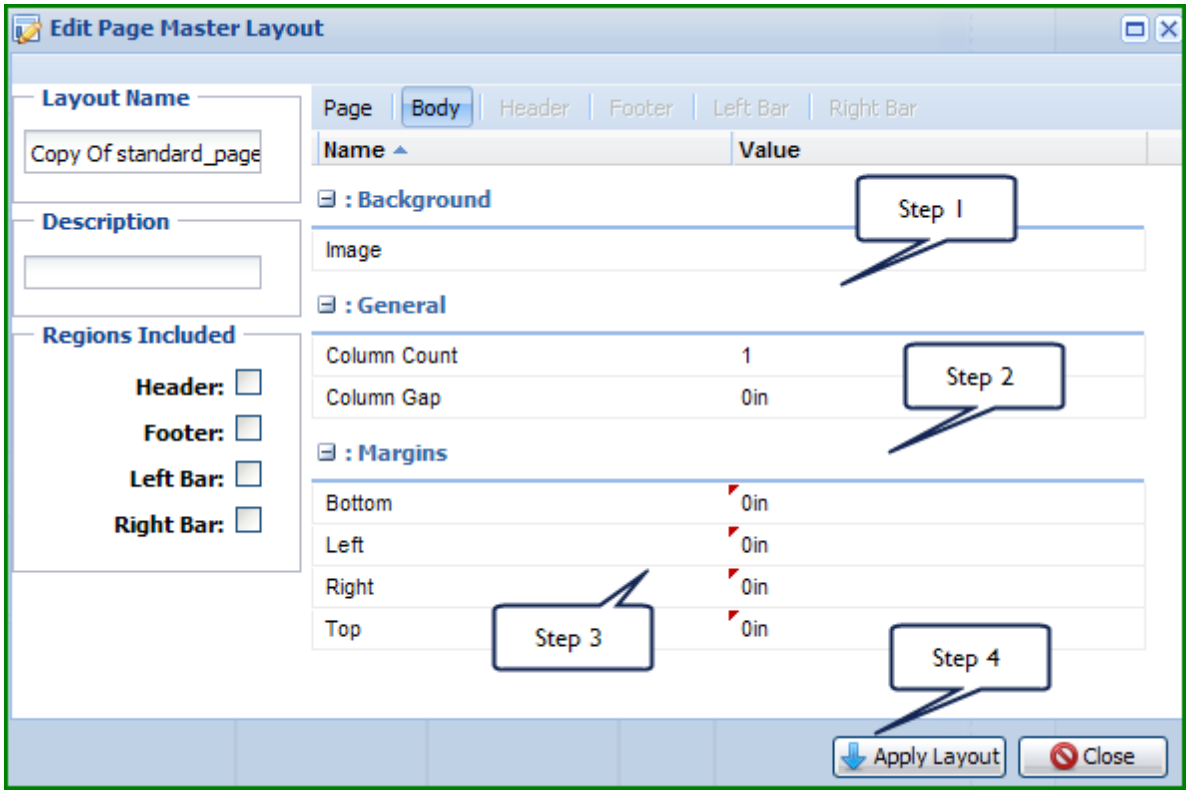


3. Click **Apply Layout** to save changes.

Body Tab

1. Specify an image location. *This image is set as the background image for the body region only.*

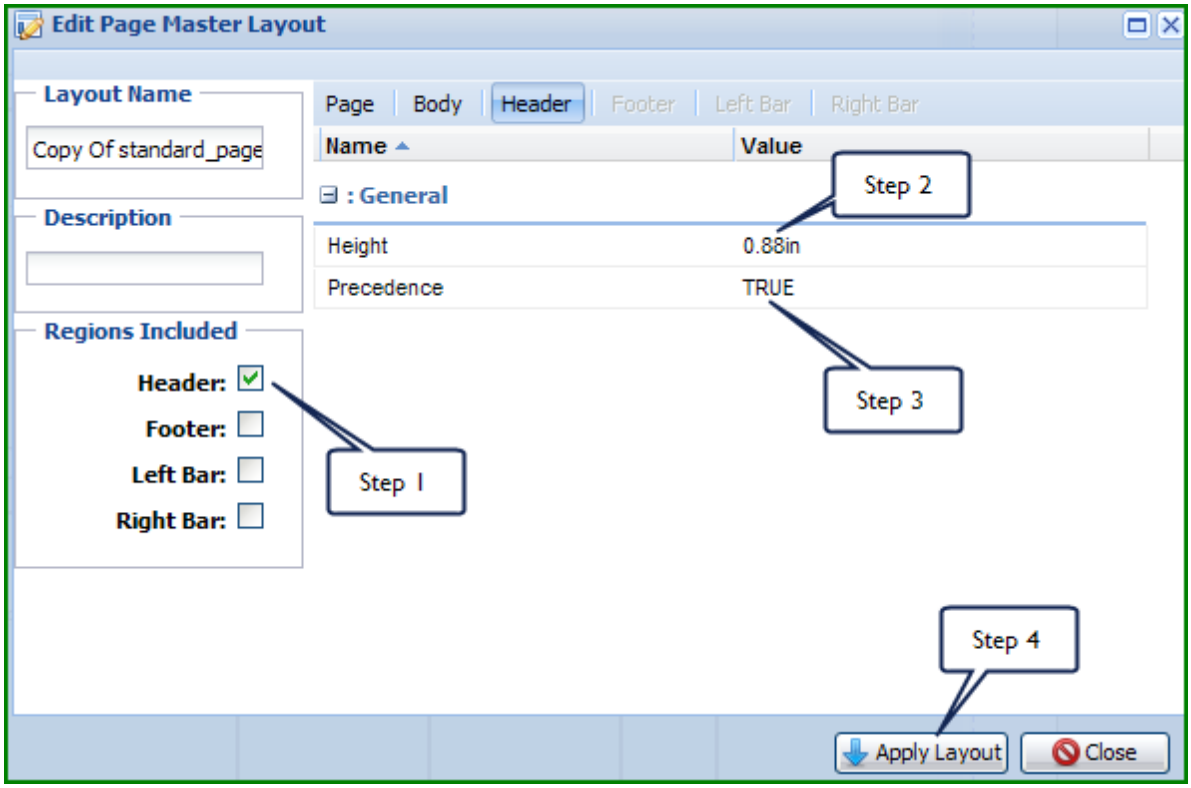
2. Specify the **Column Count** and the **Column Gap**. *The Column Count allows the user to divide the body of the page into multiple columns. The data automatically flows from one column to the other and the user cannot choose the location within a column where data can be entered. The Column Gap is the space between these columns.*



3. Set the **Margins** of the body.
4. Click **Apply Layout** to save changes.

Header Tab

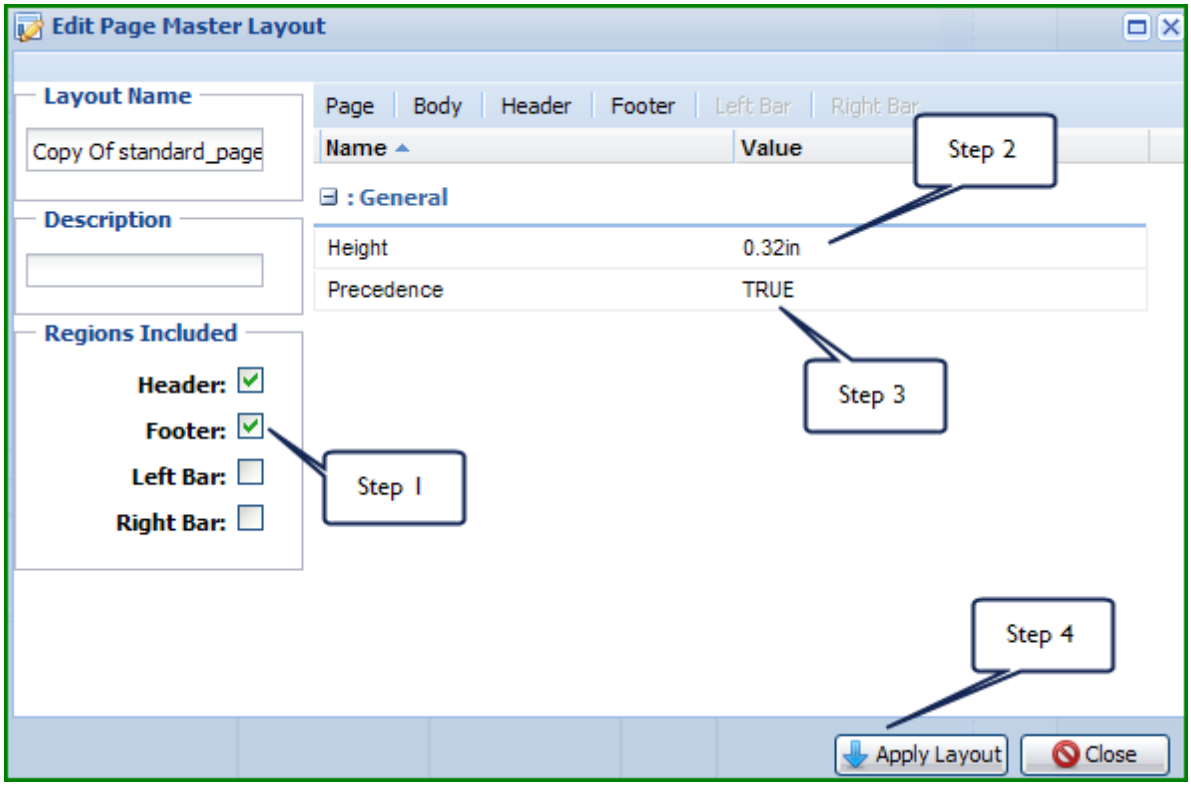
1. Select the checkbox next to "Header".
2. Enter the value to change the height of the Header.
3. Choose **True** or **False** from the drop-down menu. *Selecting True indicates that the header region extends to the corners (that is, shares space with both the side bars). Selecting False indicates that the header region does not extend to the corners.*



4. Click **Apply Layout** to save changes.

Footer Tab

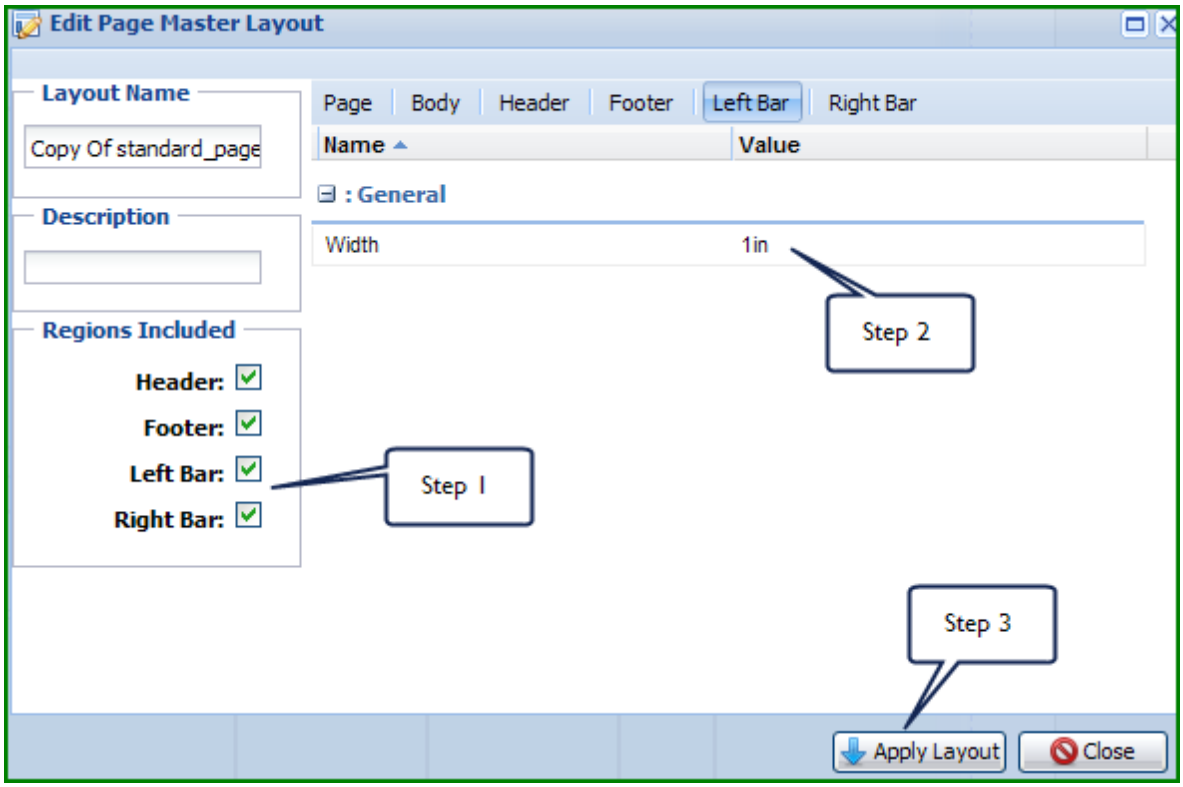
1. Select the **Footer** checkbox.
2. If desired, change the height of the **Footer**.
3. Choose **True** or **False** from the drop-down menu. *Selecting True indicates that the header region extends to the corners (that is, shares space with both the side bars). Selecting False indicates that the header region does not extend to the corners.*



4. Click **Apply Layout** to save changes.

Left & Right Bar Tab

1. Select the **Left Bar** and/or **Right Bar** checkboxes.



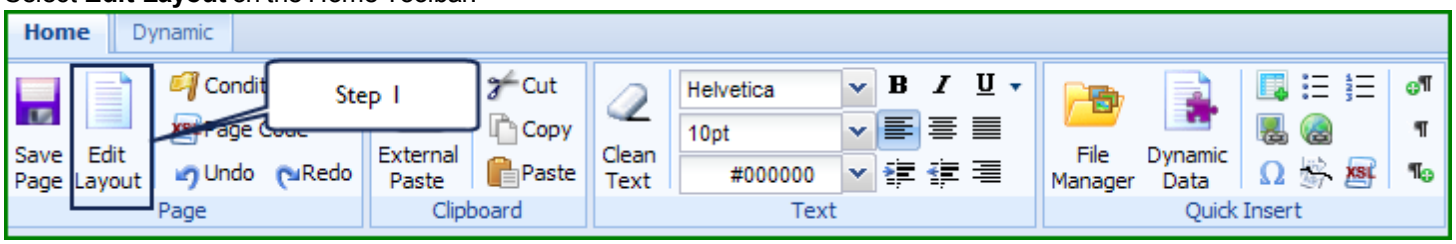
2. Enter the value of the bar's width.

3. Click **Apply Layout**.

ADMINISTRATION

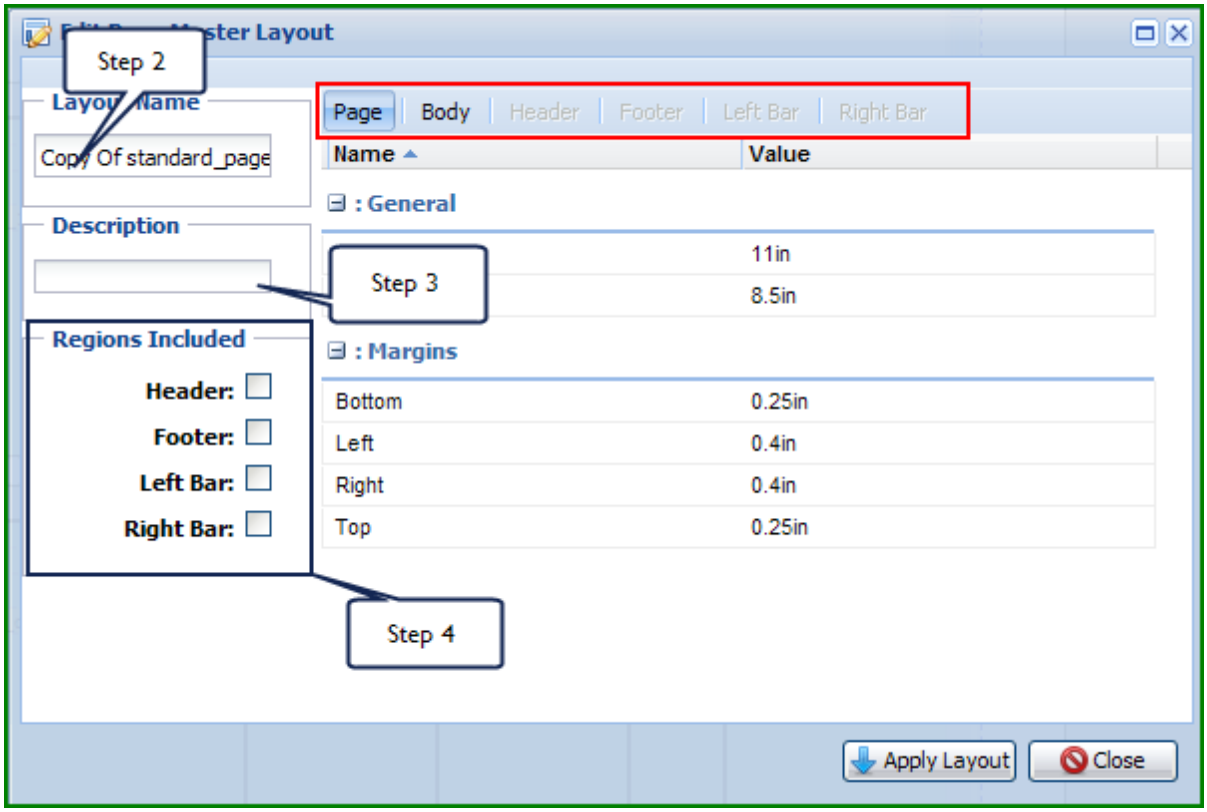
Editing the Page Layout

1. Select **Edit Layout** on the Home Toolbar.



2. Enter a **Layout Name**. *This is mandatory*.

3. Enter a valid **Description** for the layout design.



4. To enable additional regions (**Header**, **Footer**, **Left Bar** and **Right Bar**), check the box next to the specific region. *By default, the Page and Body regions are enabled for editing.*

RELATED TOPICS

EXPRESSION EDITOR

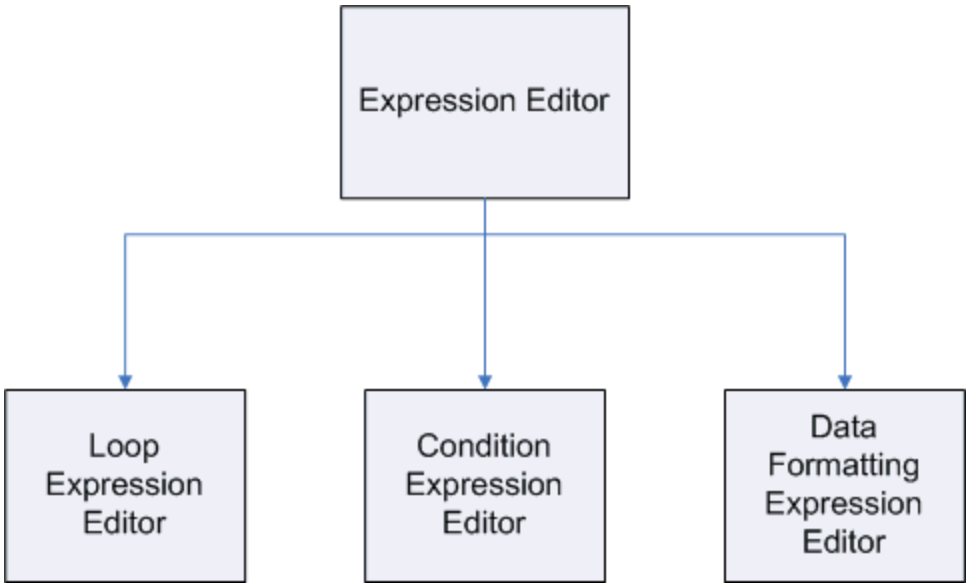
OVERVIEW

The Expression Editor is used to edit XSL expressions within a document. It allows the user to manipulate page elements using XSL functionality without having to resort to full-fledged coding.

Using the Expression Editor

Each variation of the expression editor is customized to support the specific document component.

- Common Properties:
  - **Expression Text Name** - Each Expression must have a text name.
  - **Expression Description** - The users can enter some description in this text area.
- Common Actions:
  - **Code View** - View the code that is generated by the expression. The users can directly modify the code. If code is modified, the users are prompted that saving code changes will convert this expression into custom XSL.
  - **Update** - The expression is validated and saved when this action is performed.



There are three variations of the Expression Editor:

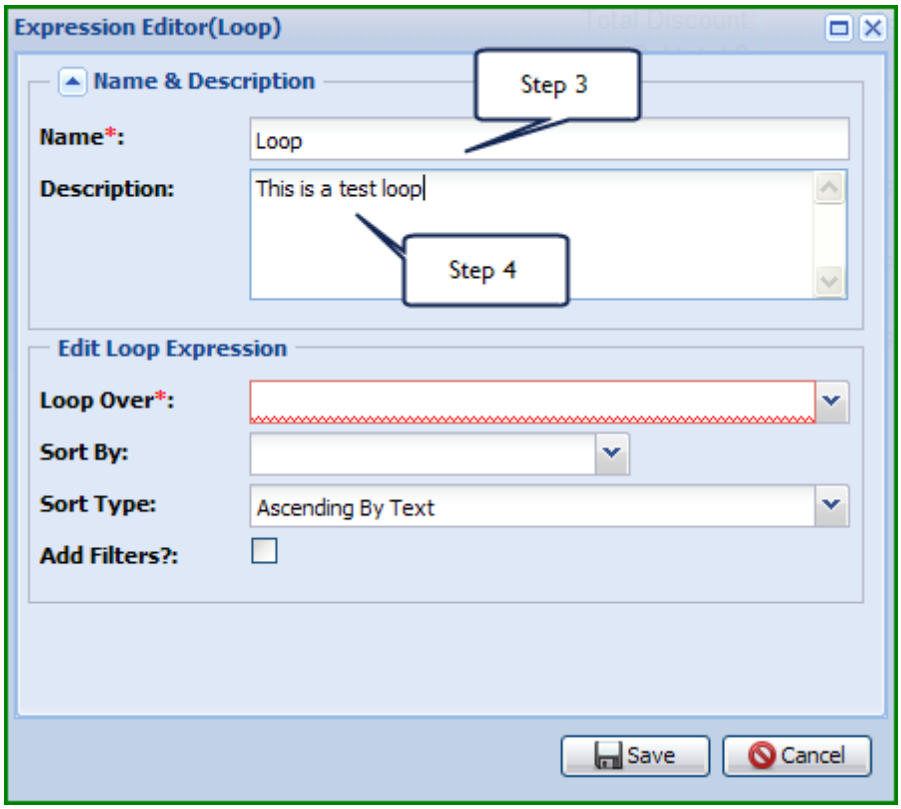
- Loop Expression Editor

Navigation: Document Editor > Dynamic Tab > Repeating Section

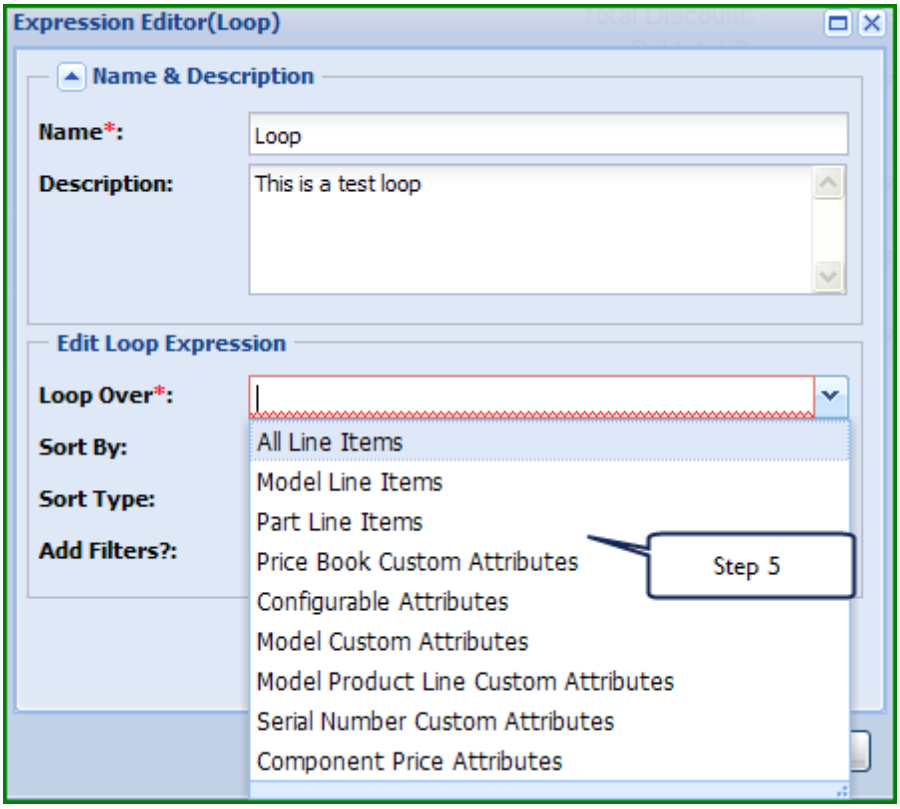
The Loop Expression Editor is used to define and edit a loop without having to resort to any coding. Loops can be edited within a page or a document.

Inserting a Loop

1. Place your cursor at a point in the Page Editor where you would like to insert the repeating section.
2. From the Dynamic Tab, select **Repeating Section** to open the Loop Expression Editor.
3. Enter a **Name** for the section.
4. Enter a **Description** for your repeating section.



5. Using the drop-down list, choose the data that is to be looped over from the drop-down menu. *Line items:*
    - **Model Line Items:** Only configured line items
    - **Part Line Items:** Only non-configured line items
    - **Price Book Custom Attributes:** Loop through the price book custom attributes (fields 1-20)
    - **Configurable Attributes:** All configurable attributes and values
    - **Model Custom Attributes:** All custom model attributes
    - **Model Product Line Custom Attributes:** All product line custom attributes
    - **Serial Number Custom Attributes:** All serial number custom attributes
    - **Component Price Attributes**



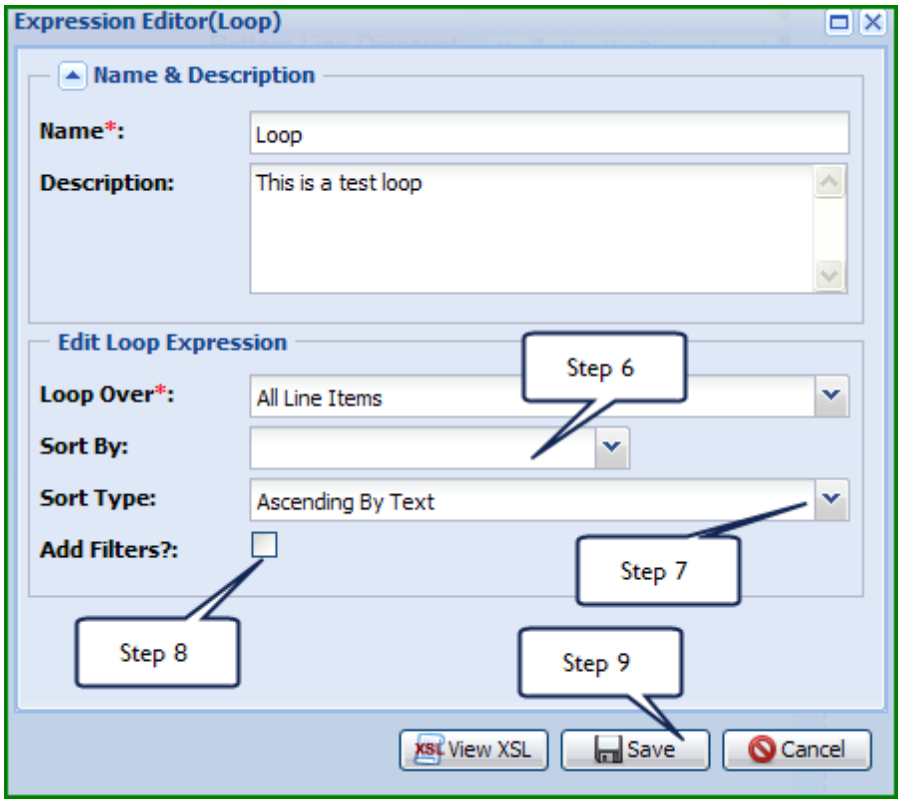
6. Using the drop-down list, choose which attribute should be used to sort the results. *The attribute that you select for **Sort By** needs to be a part of what data you are looping over.*

7. Select the method that will be used to sort based on the selected **Sort By** value. *The users can sort text or numbers in ascending order (A to Z, or zero to 9), or in descending order (Z to A, or 9 to zero).*

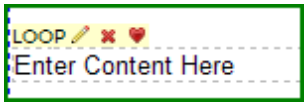
8. Select **Add Filters?** to specify the conditions that will filter out the data that the loop iterates over:

    - **Data Field:** Existing data columns that are search-able in commerce.
    - **Operator:** Enable the comparison of the data field with the value to search for and generate the filter result.
    - **Attribute Value:** Specify the value of the data field.
    - **Order of Operations:** Defines the full expression logic and logical operations between each condition attribute. Each row can be referenced by its index number assigned by the system. By default, all condition rows are AND-ed.
        - *Use And All button to AND all of the condition attributes in the order of their creation (index number).*
        - *Use the Or All button to OR all of the condition attributes in the order of their creating (index number).*
        - *You do not need to use any order for your attribute indexed while defining the order of operation. The attribute indexes can be referenced multiple times in the order of operation to define the correct condition. The order of operation defined is validated by the system when you click Update or View Code.*

9. Click **Save** to save and update the loop condition. *You will see the Loop section appear in your document, highlighted in light yellow.*



10. Insert the content to display where it reads **Enter Content Here**. *You can also insert more dynamic element (conditionals, dynamic data, loops, etc.) inside of the content.*



Click Add Row to add another row to the filter.

Editing a Loop

- **To edit a loop:** click on the Pencil icon and the Expression Editor will open. *You will see a button appear for "View XSL". Click to view the element's XSL.*
- **To delete a loop:** click on the red X.
- **To add a loop:** click on the "heart" icon.

Example Use Case: Inserting a Loop

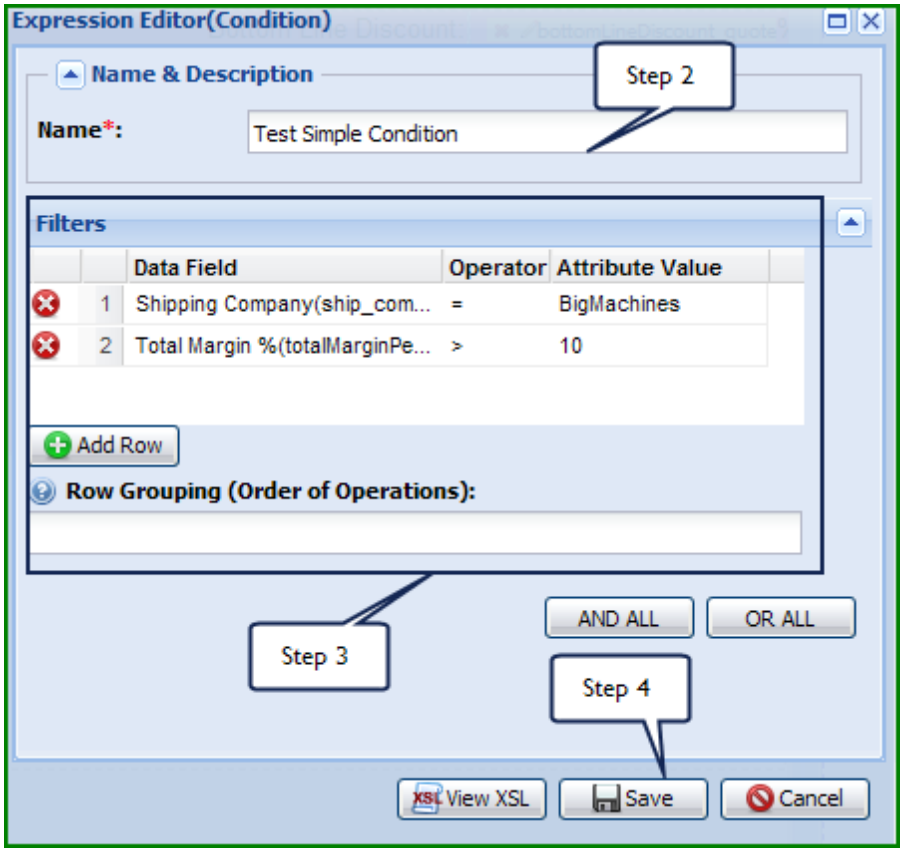
1. Output a list that contains the **Part Number** of all Part Type Line Items with Lead Time less than 5 days OR Part Base Price less than $100.
2. Specify What Data you want and how it will be displayed (build the table).
3. Create a list. *For the list element, specify a dynamic place holder for Part Number since this is the field you want to display in your list.*
4. Populate the fields you want with the right data from the Transaction (build the loop).
5. Select the list that you created above and launch the Loop Expression Editor by clicking on the **Repeating Section** action in the Dynamic Data tab.
6. Follow the steps listed above for "Inserting a Loop"
    - **For Loop Over:** Choose Part Line Items. This creates a loop that iterates through all the part type line items in your transaction.
    - **For Sort By:** Choose Part Number. This sorts the results in alphabetical order.
    - Create a filter that returns part line items with lead times less than 5 days or part base price less than $100:
        - *Choose the first data field as Lead Time. Specify that Lead Team < 5.*
        - *Choose the second data field as the User Currency Preference. Specify that Currency Preference = USD.*
        - *Choose the third data field as Part Base Price. Specify that Part Base Price < 100.*
        - *In the Row Grouping Field, group the conditions like this: (1 OR (2 AND 3)).*

- **Conditional Expression Editor**

This editor helps in specifying and editing conditions for the selected area in the page or document. If the condition is met, then the selected node(s) will be hidden in the output.

Defining a Conditional

1. Place your cursor at a point in the page where you would like to insert a conditional.
2. Enter a **Name** and **Description** for the condition.

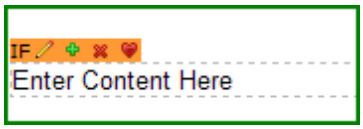


3. Add filters to specify the conditions. *Complete the following information:*

    - **Data Field:** Select from the existing data columns that are search-able in commerce.
    - **Operator:** Select from the list of operators in the field by clicking directly on the field. The operators enable the comparison of the data field with the value to search for and generate the filter result.
    - **Attribute Value:** Specify the value of the data field.
    - **Order of Operations:** Use this section to define the full expression logic and logical operations between each condition attribute. Each row can be referenced by its index number assigned by the system. By default, all condition rows are AND-ed.
        - *Use And All button to AND all of the condition attributes in the order of their creation (index number).*
        - *Use the Or All button to OR all of the condition attributes in the order of their creating (index number).*
        - *You do not need to use any order for your attribute indexed while defining the order of operation. The attribute indexes can be referenced multiple times in the order of operation to define the correct condition. The order of operation defined is validated by the system when you click Update or View Code.*

4. Click **Save** to save changes. *If this is a new condition, it will now appear on your document highlighted in Orange. Click Cancel to exit the Expression Editor without saving changes.*

5. In the IF element, insert the contents where it says, "Enter Contents Here". *You can also insert more dynamic elements (conditionals, dynamic data, repeating section, etc.) inside of the content.*



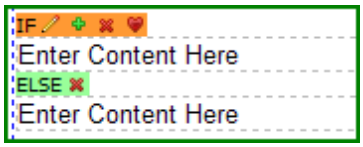
Click Add Row to add another row to the filter.

Adding an ELSE Clause

1. In the conditional "IF" section icon, double-click on the GREEN "+" button. *An ELSE block should appear below the "IF" content.*



2. Edit the content in the dotted box which will display if the condition in the "IF" is not met.



If you need to remove the ELSE block, simply click the "X" icon next to the ELSE label. *This will remove all content inside of the ELSE block.*

Example: Conditional Formatting

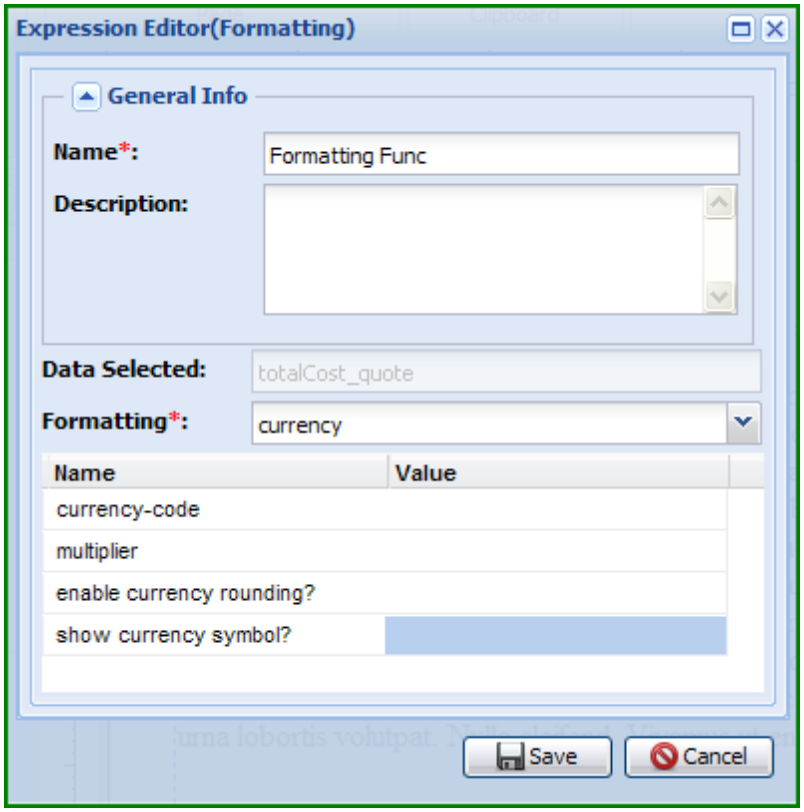
Show a part in the contract named "For California Buyers" to buyers from California only. Buyers from other states should not see this section in the contract.

1. Name the section: "For California Buyers" - use the rich text editor to author the content.
2. Hide the section: Select the "For California Buyers" text and launch the Conditional Expression Editor. In the Data Fields, choose the CRM Attribute "Buying State Province".
3. Specify the condition "Buying State Province = California".

When a buyer, who is not from the state of California logs in, the condition is met and the section is hidden. For a buyer from California, the condition is false and the text is therefore no longer hidden.

- **Data Formatting Expression Editor**

The Data Formatting Expression Editor is used to define and edit the format for dynamic data without having to resort to any coding.



Defining a Format

- **Source:** This indicates that the dynamic commerce data will be derived from the current process.
- **Name:** Enter a name for the dynamic data that you want to format. This is a required field. The text name is not unique.
- **Description:** Enter a valid description for the data to be formatted.
- **Data Selected:** This is a read-only field that indicates that the dynamic data field was selected.
- **Formatting:** Use the tree drop-down list to select the function(type) to be applied to the data. This function will be applied to the dynamic data from the commerce XML. The result of applying the function on the data is displayed on the output.

- **Update:** Click Update to update the loop condition in the database.
- **Cancel:** Click Cancel to cancel the loop condition definition.
- **Parameter Grid:** This section displays parameters of the formatting function. The number of parameters and the type of parameter depends on the function that is chosen in the Formatting section.
    - ***Name:*** *Provides a name for the parameter that is expected by the function picked in the Formatting field*
    - ***Value:*** *Edit the values of the parameters in the Value field.*

Currency Formatting

We support dynamic values for currency-code and value multiplier in printed documents. There are additional parameters that control rounding and display of currency symbol. Multi-currency applications can take advantage of this feature to print price and discount attributes in any supported currency regardless of the currency of the current user of the currency in which the document was initially created.

Currency Function Parameter	Description
Currency_code	The fullaccessuser can now pass a global XSL variable name in this field so that currency is shown depending on the logic specified by the fullaccessuser. The default value for this field is the current user's preferred currency.
Multiplier	The fullaccessuser can now pass a global XSL variable name in this field so that the correct multiplier is applied on the logic specified by the fullaccessuser in the global XSL snippet. The default value for this field is one.
Enable Currency Rounding	When enabled, this option will round the currency value. Be default, this option is set to false.
Show Currency Symbol	When enabled, this option will show the currency symbol depending on the currency specified in the currency-code parameter. Be default, this option is set to true.

Example: Data Formatting

Goal: The cover letter must be addressed to the Name of the Buyer. The name of the buyer is the user name contained in the email address. For example, if the user's email is dvader@empire.com, then the letter must be addressed to dvader.

Solution:

- *Specify what data you want:* Drag and Drop the **User Email Dynamic Data** at the selected node in the document page.
- *Format your data:*
    - In the **Formatting** section, choose substring-before.
    - In the **Value** section, specify "@". This returns the string before the "@". In our example, when the email is dvader-@empire.com, it returns dvader.

ADMINISTRATION

Formatting Expression Editor Parameters: When defining a format in the Formatting Expression Editor, use the tree drop-down list to select the function(type) to be applied to the data. This function will be applied to the dynamic data from the commerce XML. The result of applying the function on the data is displayed on the output.

General Functions

Function	Parameters	Description
normal-size	None	Trims any extra whitespace before and after the data value.
count	None	Returns the number of times the data value occurs.
sum	None	Returns the sum of the numerical value of each node in the data value.

String Functions

Function	Parameters	Description
substring	**start-index:** {number} starting position to take the substring from; **length:** {number} length of string to take	The substring function returns the substring of the data value starting at the start-index with length of length. For example, if the data value is "12345", then substring(2,3) returns "234".
substring-after	search string: {text}: starting position to take the substring from	The substring-after function returns the substring of the data value that follows the first occurrence of the search string, or the empty string if the search string was not found. For example, if the data value is "1999/04/01", the substring after("/") returns 04/01 and substring after("19") returns 99/04/01.
substring-before	search string: {text}: starting position to take the substring from	The substring-before function returns the substring of the data value that precedes the first occurrence of the search string, or the empty string if the search string was not found. For example, if the data value is "1999/04/01", then substring before("/") returns 1999.
concat	param 1: {text} string to concatenate to end of a data value	The concat function returns the concatenation of its arguments.
string-length	None	The string-length function returns the number of characters in the data value.
translate	**search-characters:** {text} starting position to take the substring from; **replace-characters:** {text} length of string to take	The translate function returns the data value with occurrences of characters in the search-characters parameter replaced by the character in the replace-character parameters. **For example:** if the data value is "bar", then translate("abc", "ABC") returns the string "BAr". If there is a character in the search-characters parameter with no character at a corresponding position in the replace-characters parameter, then occurrences of that character in the first argument string are removed. For example, translate("abc-", "ABC"), where the data value "---aaa--", returns "AAA". *If a character occurs more than once in the search-characters parameter, then the first occurrence determines the replacement character. If the replace-characters parameter is longer than the search-characters parameter, then excess characters are ignored.*

Numeric Functions

Most of these functions are standard XSLT functions. Detailed documentation on parameters can be found on XSLT reference web sites such as: http://www.devguru.com/Technologies/xslt/quickref/xslt_index_functions

Function	Parameters	Description
format-number	**format-number:** {text} *Required*- Specifies the format pattern; **decimal-format:** {text} *Optional*	The format-number function is used to convert the data value into a formatted string. Characters used in the formatting pattern: **#** - Denotes a digit; **0** - Denotes leading and following zeroes; **.** - Denotes a decimal point; **,** - The group separator for thousands; **%** - Displays the number as a percentage; **;** - Pattern separator. The first pattern will be used for positive numbers and the second for negative numbers.
currency	currency-code: {text} *Required*	Formats a number into a currency string.
round	None	Rounds the data to the nearest integer.
floor	None	Returns the largest integer that is not greater than the data value.
ceiling	None	Returns the smallest integer that is greater than the data value.

Date Functions

There are no predefined functions available to format dates as **dd.MM.yyyy** or **dd/MM/yyyy**. The fullaccessuser will have to write a custom XSL to achieve this formatting.

Function	Parameters	Description
shortdate	separator: {text} *Required*. The delimiter to use, "/" in the example to the right.	Formats a BigMachines date attribute into a short-hand readable format such as 2/12/2008.
longdate	separator: {text} *Required though not used.* Enter anything into this field.	Formats a BigMachines date attribute into a long-hand readable format, such as February 12, 2008.

RELATED TOPICS

COMPLEX CONDITIONALS

OVERVIEW

The Complex Conditionals functionality provides the ability to insert complex condition statements with multiple if...else clauses inline within the content of a document using the Document Editor without having to resort to custom XSL snippets. The fullaccessuser can create complex conditionals within the Complex Conditional Expression Editor. The `<if...then>` conditional statements can be set up using simple filters. The content within a conditional can be created in a WSYWYG editor. All editor functionality such as formatting, text, hyperlinks, images, tables, dynamic data, repeating sections and conditionals are available within this editor. However, the user is not allowed to insert a complex conditional within a complex conditional.

The page code reflect the conditional in the following format:

```
<xsl:call-template name="Condition1"><  xsl:call-template>
```

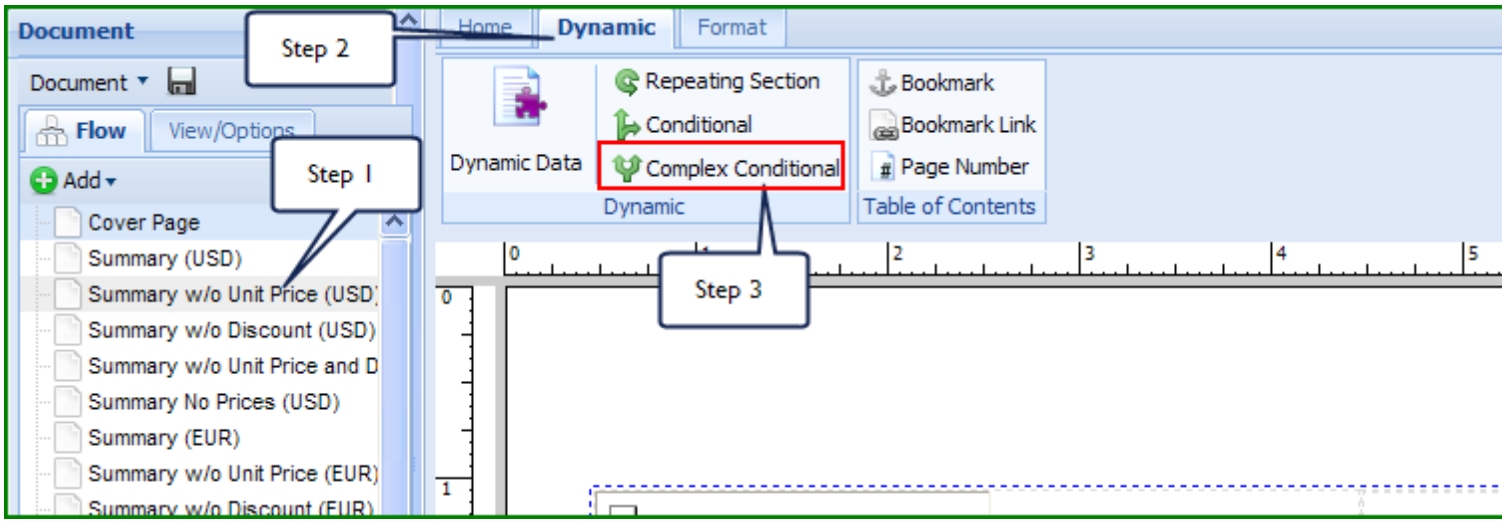
Example Use Case: In the case of headers and footers, if you have to make a change to a logo or perhaps office lcoation information, you can simply make the change in the complex conditional and the information will update in all places that the complex conditional is located.

ADMINISTRATION

Adding a Complex Conditional

The fullaccessuser can create complex conditional statements within the Complex Conditional Expression Editor. The `<if...else>` statements can be Always True or they can be set-up using simple filters.

1. Double-click on the page you wish to edit.
2. Select the **Dynamic** tab in the toolbar.
3. Select **Complex Conditionals** to open the Complex Conditional Expression Editor dialog.



4. Enter a unique name for the complex conditional.

5. Choose a Category from the drop-down menu. *If no Categories are available, or the appropriate Category is unavailable, type a new category name directly into the text field.*

6. If this complex conditional is Always True, select the box next to Always True (this will hide the filters section). *Once it is saved as Always True, it cannot be changed.*

7. If the complex conditional is NOT Always True, you will need to create simple filters:

    - **Data Field:** Select from the exiting data columns that are in commerce. Click on the field and select from the single-select drop-down menu.

    - **Operator:** Select from the list of operators in the field by clicking on this field. The operators enable comparing the data field with the Attribute Value to search for, and generate the filter result.

    - **Attribute Value:** Specify the value of the data field to filter in this field.

    - **Add Row:** If you need to add additional rows to filter, click Add Row.

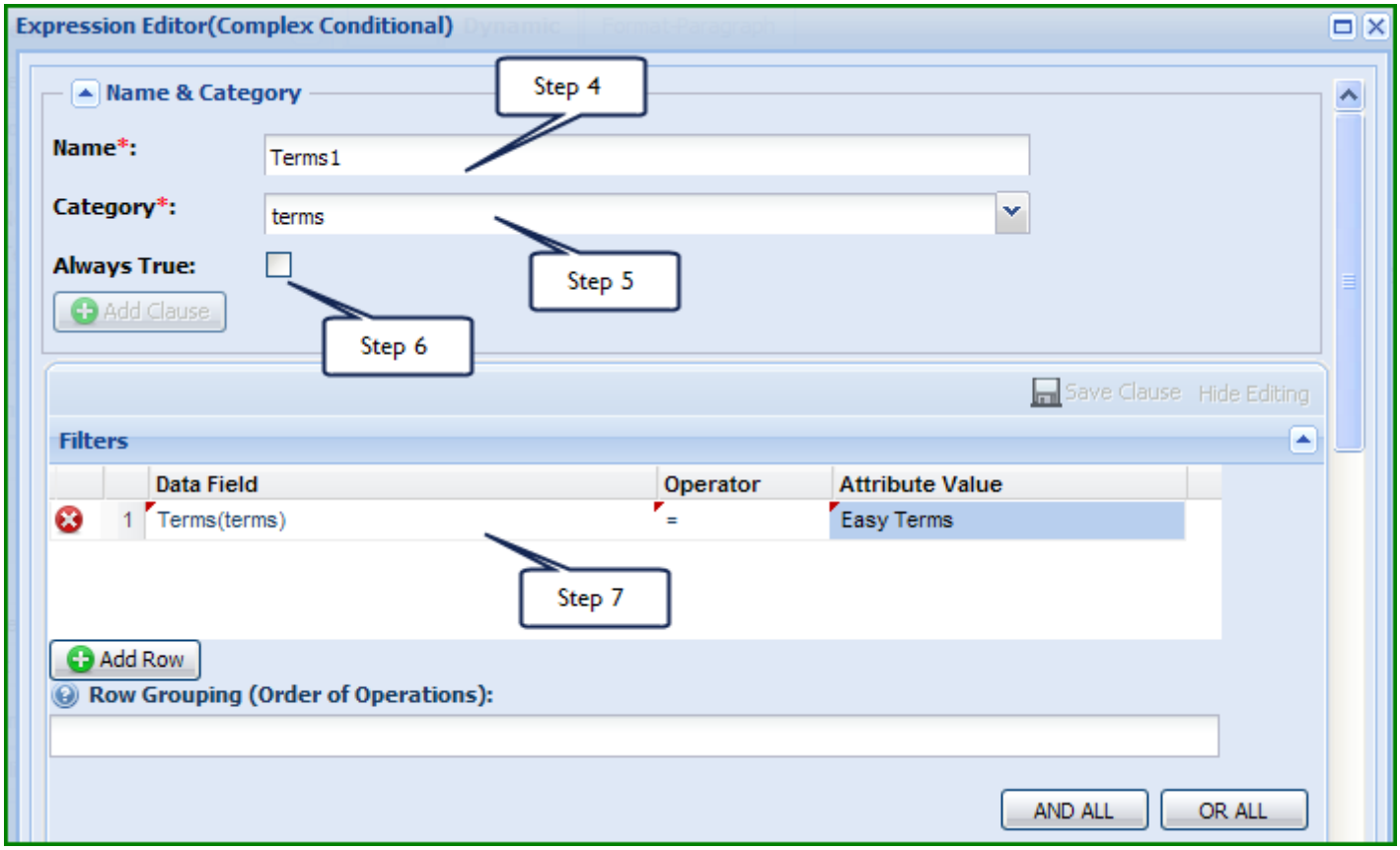
    - **Row Grouping (Order of Operations):** Use this section to define the full expression logic and logical operations between each attribute. Each row can be referenced by its index number assigned by the system. By default, all condition rows are AND-ed.

        - Use the AND ALL button to AND all the attributes in the order of their creation (index number).

        - Use the OR ALL button to OR all the attributes in the order of their creation.

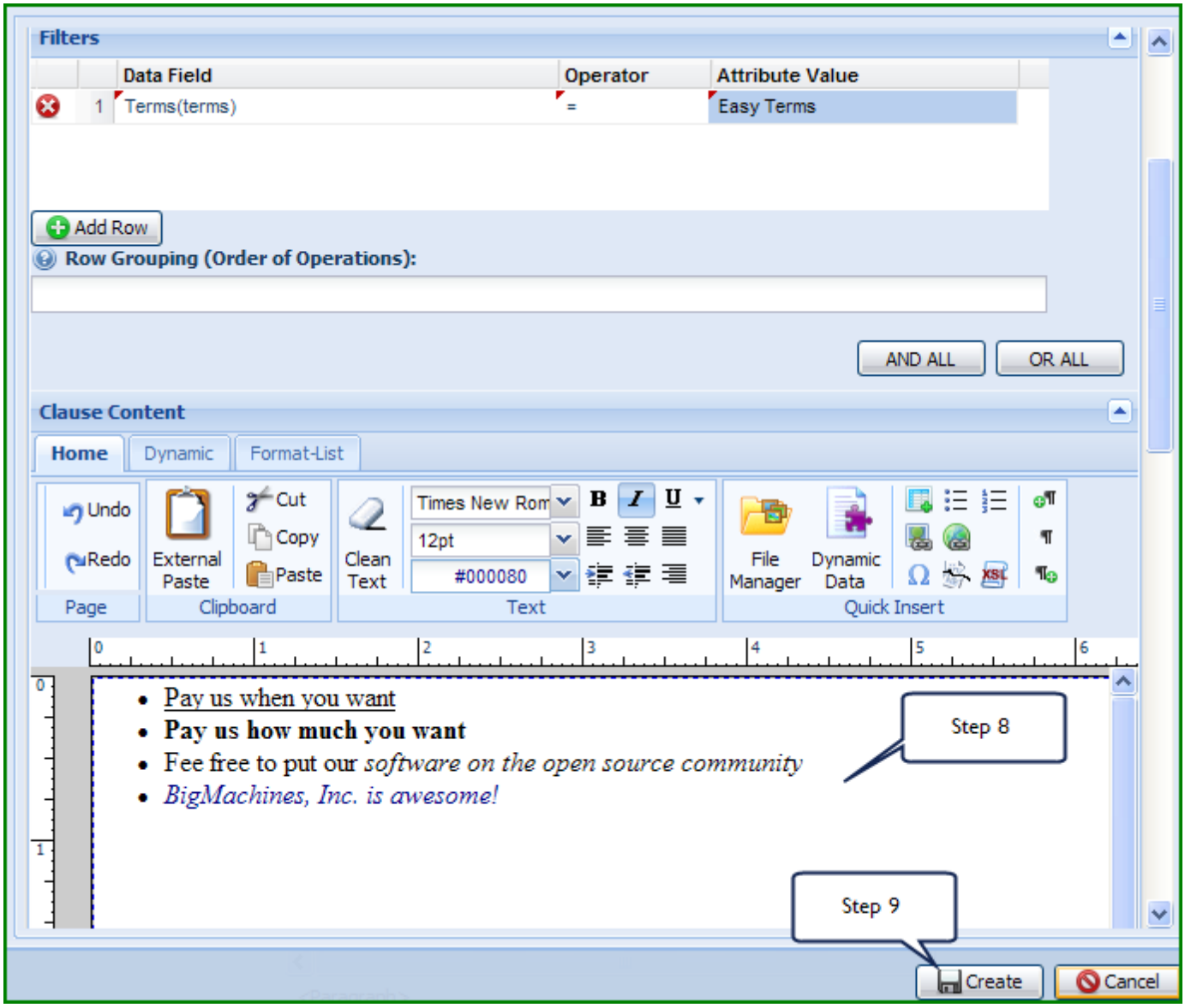
        - You can use parentheses to further control the order of operations.

        - You do not need to use any order for your attribute indexed while defining the order of operations. The attribute indexes can be referenced multiple times in the order of operations to define the correct condition. The order of operation defined is validated by the system when you click Create.

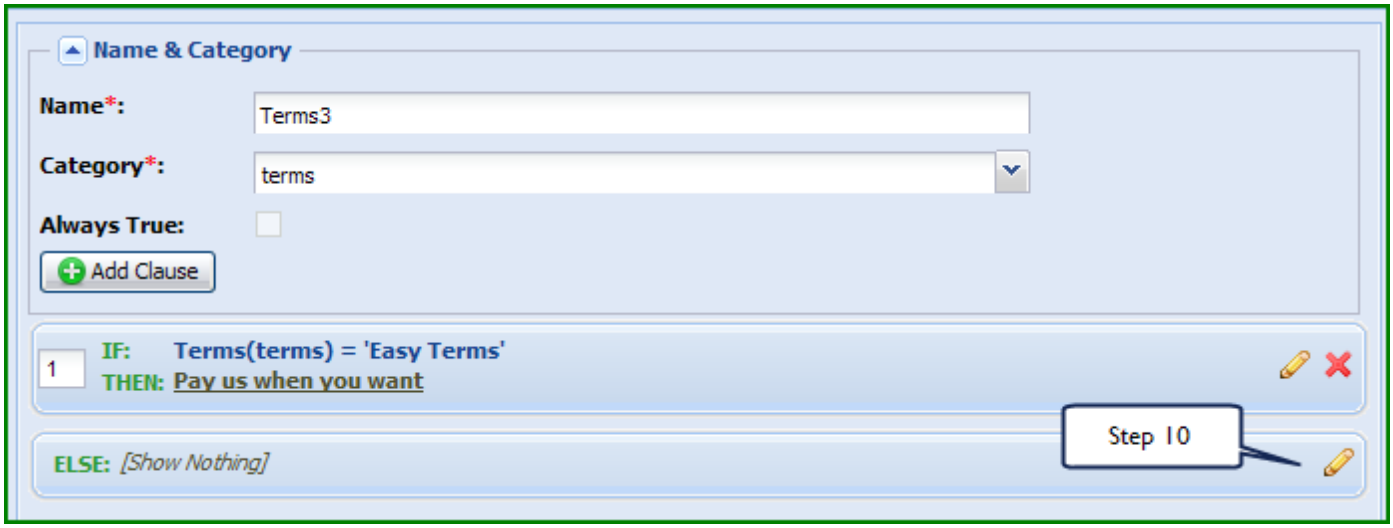


8. Format text, insert hyperlinks, images, tables, dynamic data, repeating sections, and conditionals within the Clause Content section. *This functions similarly to the WYSIWYG editor used throughout Document Engine.* You CANNOT insert a complex conditional.

9. Click **Create**.

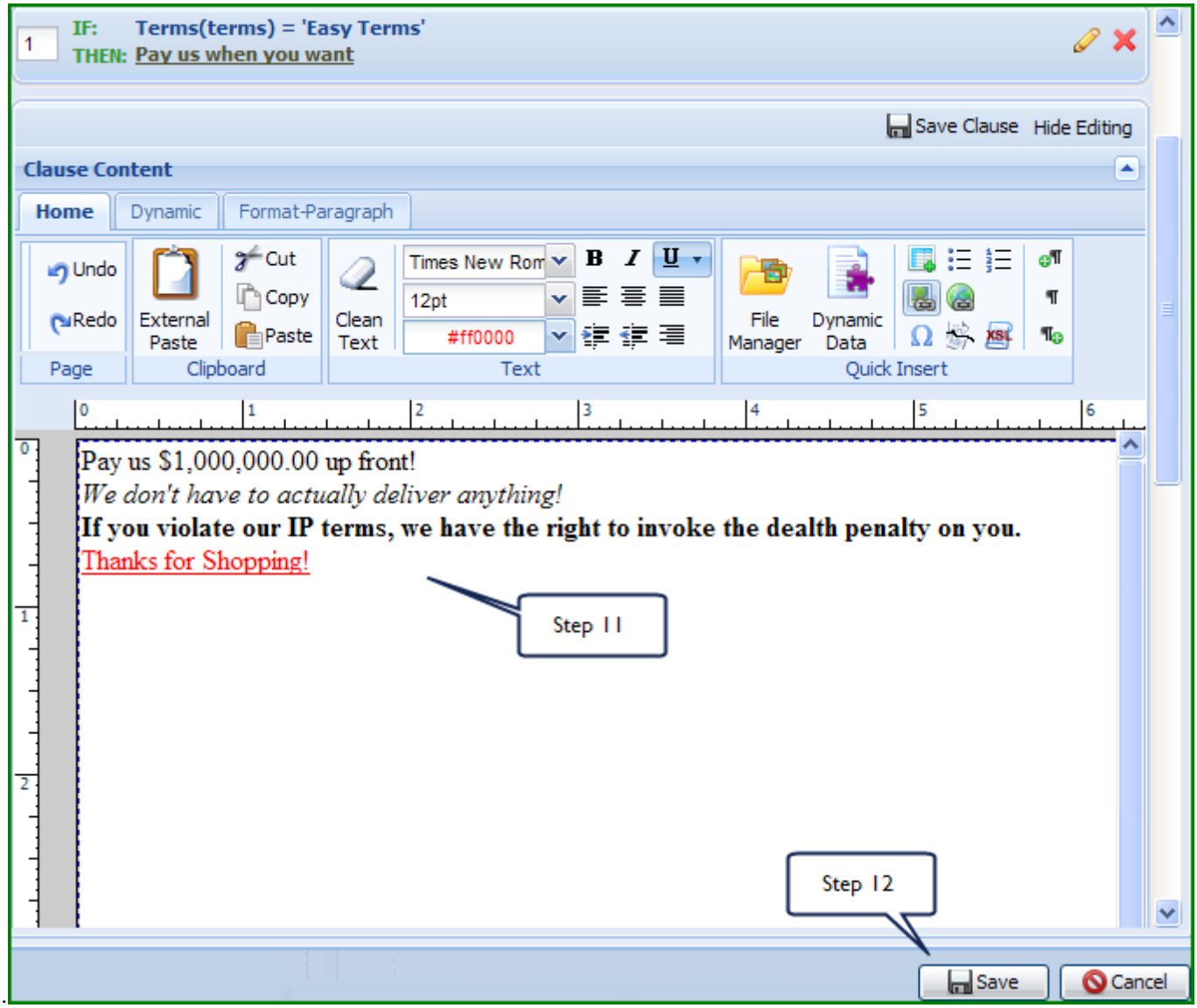


10. Click the pencil icon to add the Clause Content for the "Else" statement.



11. Format text, insert hyperlinks, images, tables, dynamic data, repeating sections, and conditionals within the Clause Content section. *This functions similarly to the WYSIWYG editor used throughout Document Engine.* You CANNOT insert a complex conditional.

12. Click **Save**



Clauses within a complex conditionals can be reordered after creation.

Conditional sections that contain a lot of data or that are nested tables/loops/conditions should be replaced with complex conditionals.

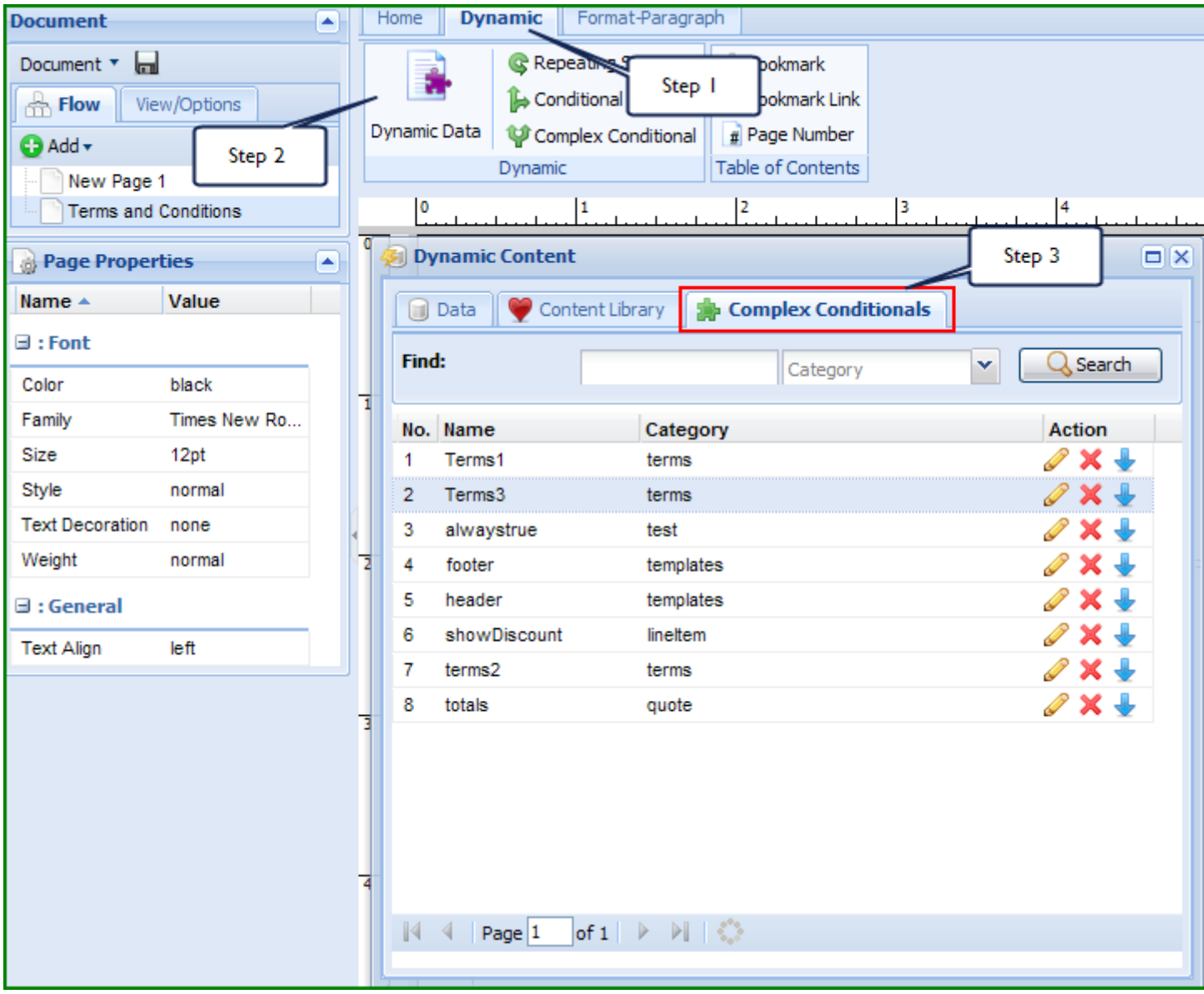
It is Best Practice to store headers, footer, terms and conditions in complex conditionals.

Complex Conditional Library

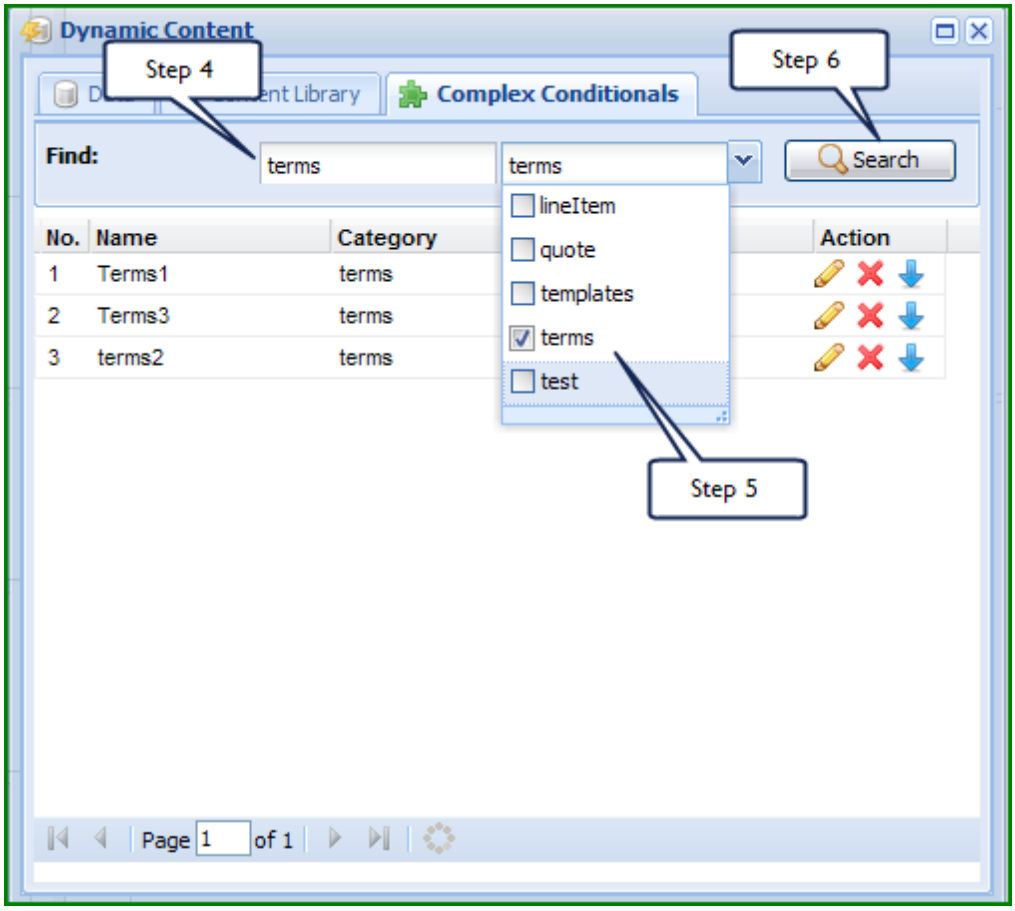
The Complex Conditionals functionality provides the ability to insert complex condition statements with multiple `if...else` clauses inline within the content of a document using the Document Editor without having to resort to custom XSL snippets. Once you have created a complex conditional, it is then stored in the Complex Conditional Library which is found when you navigate to Dynamic Data.

Using the Complex Conditional Library

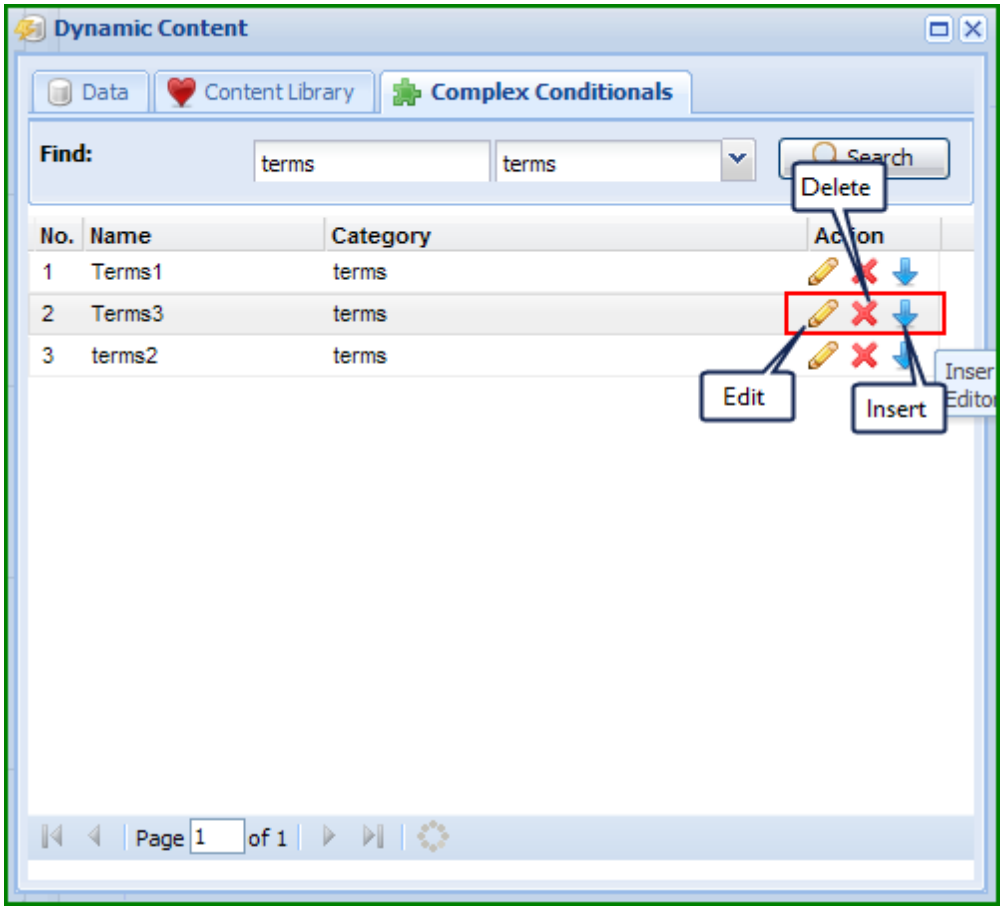
1. From any page in the Document Engine, select the **Dynamic Tab**.
2. Click on the Dynamic Data icon. *You can also access this from the Home Toolbar.*
3. Select the **Complex Conditionals** tab in the Dynamic Content dialog.



4. Search the library for your complex conditional by entering the entire name or part of the name in the text field next to **Find**.
5. Search the library by narrowing down the complex conditionals by category.
6. Click **Search** to list all complex conditionals that meet your search requirements.

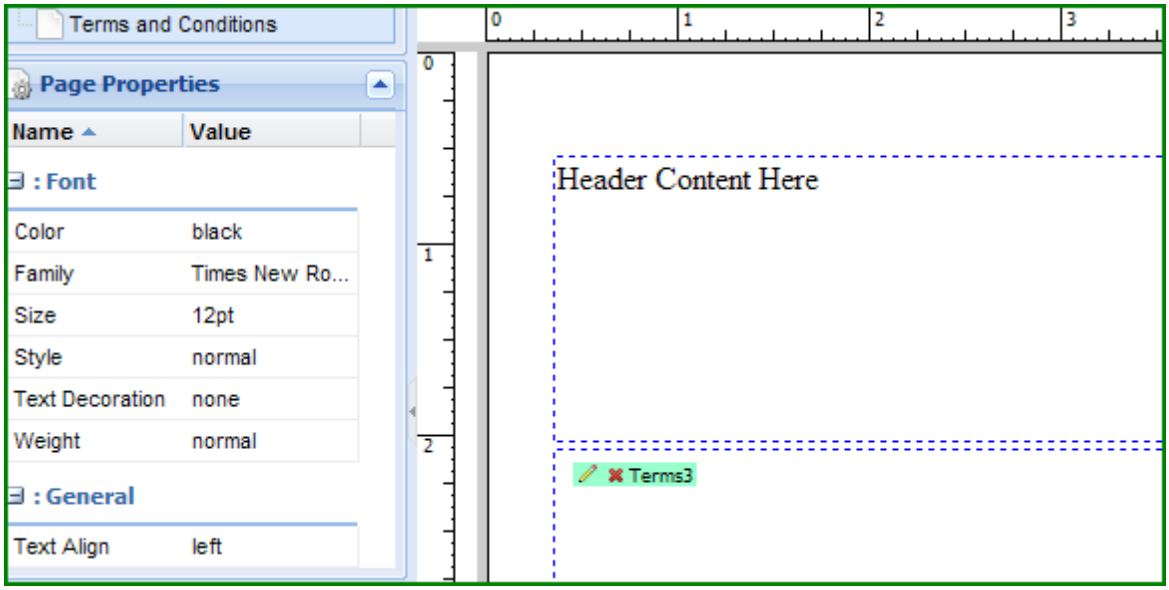


7. Select the complex conditional you search for. *You can either edit, delete or insert this into your page.*



Inserting a Complex Conditional

1. Place your cursor on the page, where you want to insert the complex conditional.
2. Navigate to the Complex Conditionals library and insert a complex conditional into your page by clicking the blue **Insert** arrow.



You will notice that the complex conditional is shown in green. You can edit this by clicking the pencil or delete it from your page by clicking the red "x".

Placing Tables back into Conditionals

If a Table somehow ends up residing outside of the Conditionals they were created in, follow these steps to easily replace Tables within Conditionals. *This allows you to simply recreate the Conditional rather than the entire Table.*

1. Click within the Table that has slipped outside of the Conditional. *The level at which you are editing will appear at the bottom of the Doc Engine.*
2. Click the ”<Table>” button to select the entire table.
3. Select the **Dynamic** tab.
4. Click the **Conditional** button.
5. Enter the conditions under which you'd like the table to appear.

Tables cannot be copied/pasted.

NOTES



Page Number within a Complex Conditional - Page numbers can be dynamically inserted into the content of a complex conditional. The value of the page number within a complex conditional reflect the page number of the current page of the document.

Clauses within a complex conditional can be reordered after creation. Saved Complex Conditionals can be used in all documents within a commerce process.



Content has to be saved once before the user is able to use dynamic data, repeating sections and rows in the Content Editor.

Complex conditionals are shared globally across all documents.

An Always True Complex Conditional cannot be changed to one with multiple conditions after creation - A complex conditional with one or more conditionals can be changed into an Always True type complex conditional. However, a complex conditional that has once been saved as Always True cannot be changed to one with one or more conditional blocks. It is therefore recommended that the fullaccessuser only use the Always True option when certain that no conditionals will be required.

RELATED TOPICS

COMPLEX CONDITIONALS

OVERVIEW

The Complex Conditionals functionality provides the ability to insert complex condition statements with multiple if...else clauses inline within the content of a document using the Document Editor without having to resort to custom XSL snippets. The fullaccessuser can create complex conditionals within the Complex Conditional Expression Editor. The `<if...then>` conditional statements can be set up using simple filters. The content within a conditional can be created in a WSYWYG editor. All editor functionality such as formatting, text, hyperlinks, images, tables, dynamic data, repeating sections and conditionals are available within this editor. However, the user is not allowed to insert a complex conditional within a complex conditional.

The page code reflect the conditional in the following format:

```
<xsl:call-template name="Condition1"><  xsl:call-template>
```

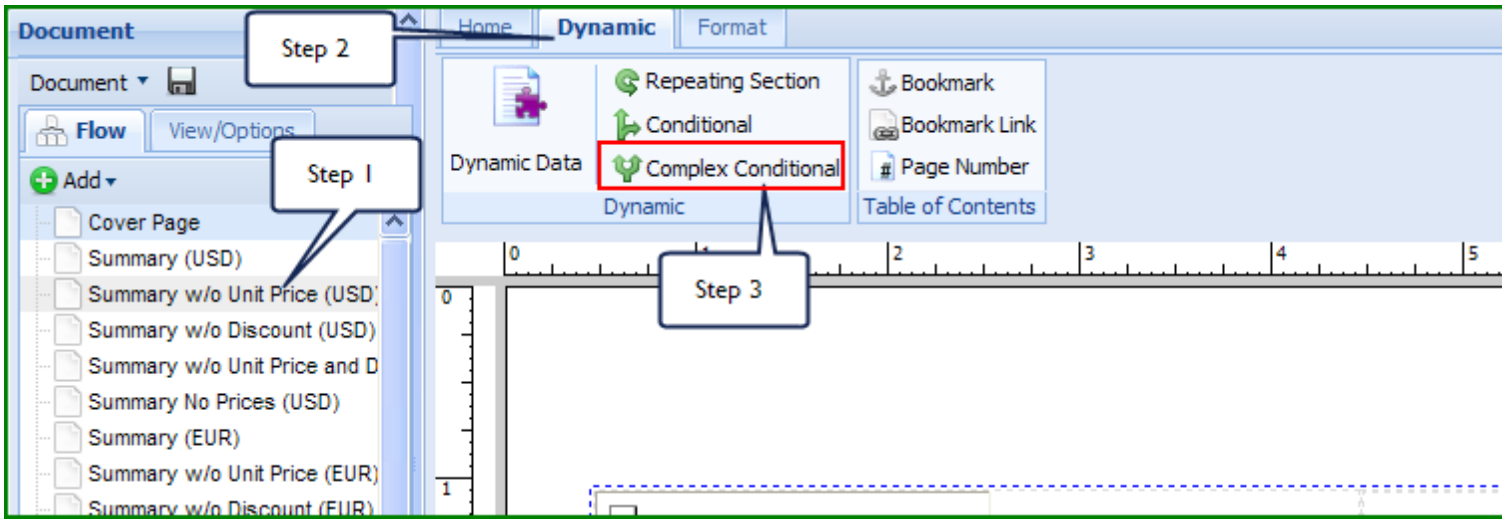
Example Use Case: In the case of headers and footers, if you have to make a change to a logo or perhaps office lcoation information, you can simply make the change in the complex conditional and the information will update in all places that the complex conditional is located.

ADMINISTRATION

Adding a Complex Conditional

The fullaccessuser can create complex conditional statements within the Complex Conditional Expression Editor. The `<if...else>` statements can be Always True or they can be set-up using simple filters.

1. Double-click on the page you wish to edit.
2. Select the **Dynamic** tab in the toolbar.
3. Select **Complex Conditionals** to open the Complex Conditional Expression Editor dialog.



4. Enter a unique name for the complex conditional.

5. Choose a Category from the drop-down menu. *If no Categories are available, or the appropriate Category is unavailable, type a new category name directly into the text field.*

6. If this complex conditional is Always True, select the box next to Always True (this will hide the filters section). *Once it is saved as Always True, it cannot be changed.*

7. If the complex conditional is NOT Always True, you will need to create simple filters:

    - **Data Field:** Select from the exiting data columns that are in commerce. Click on the field and select from the single-select drop-down menu.

    - **Operator:** Select from the list of operators in the field by clicking on this field. The operators enable comparing the data field with the Attribute Value to search for, and generate the filter result.

    - **Attribute Value:** Specify the value of the data field to filter in this field.

    - **Add Row:** If you need to add additional rows to filter, click Add Row.

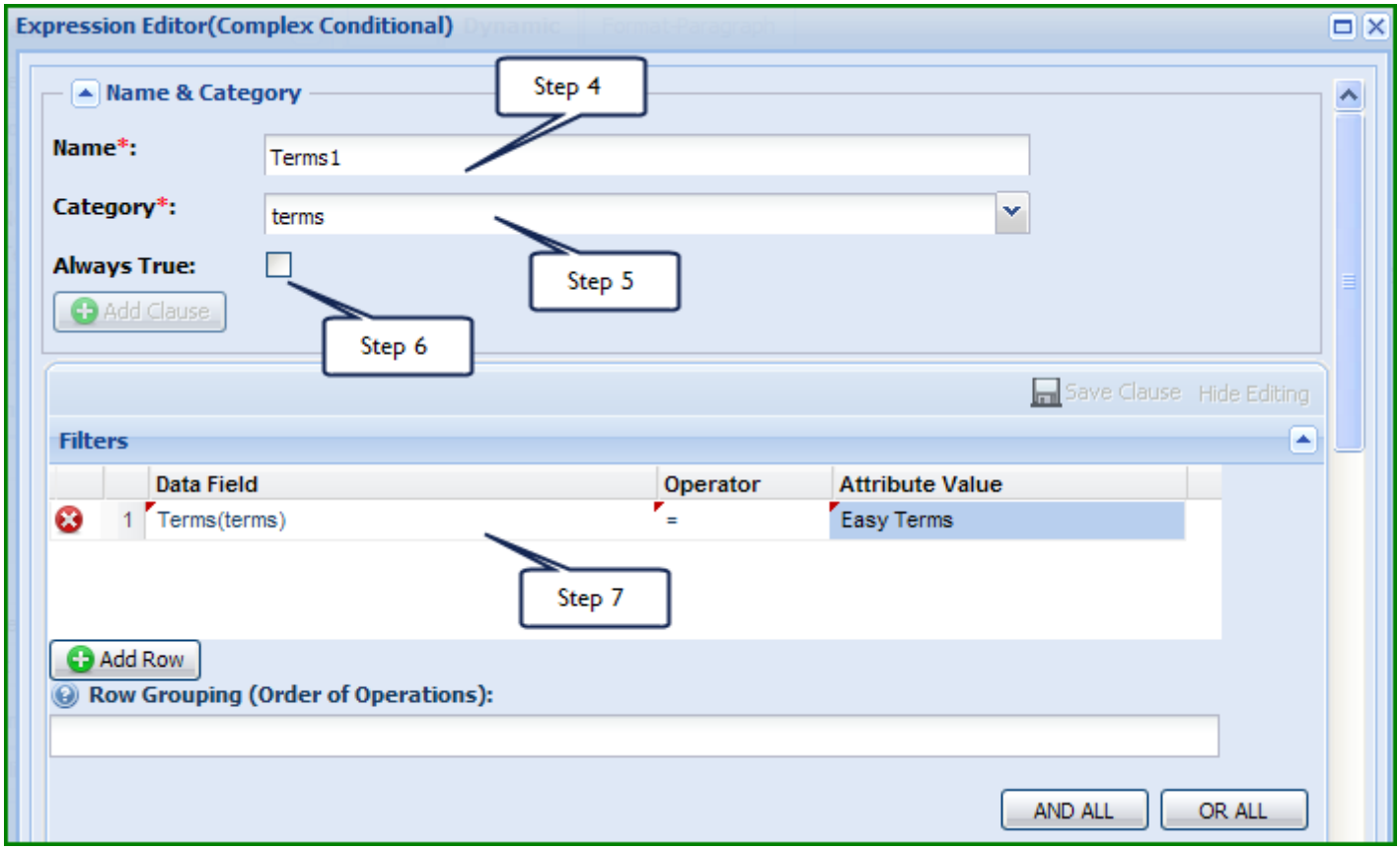
    - **Row Grouping (Order of Operations):** Use this section to define the full expression logic and logical operations between each attribute. Each row can be referenced by its index number assigned by the system. By default, all condition rows are AND-ed.

        - Use the AND ALL button to AND all the attributes in the order of their creation (index number).

        - Use the OR ALL button to OR all the attributes in the order of their creation.

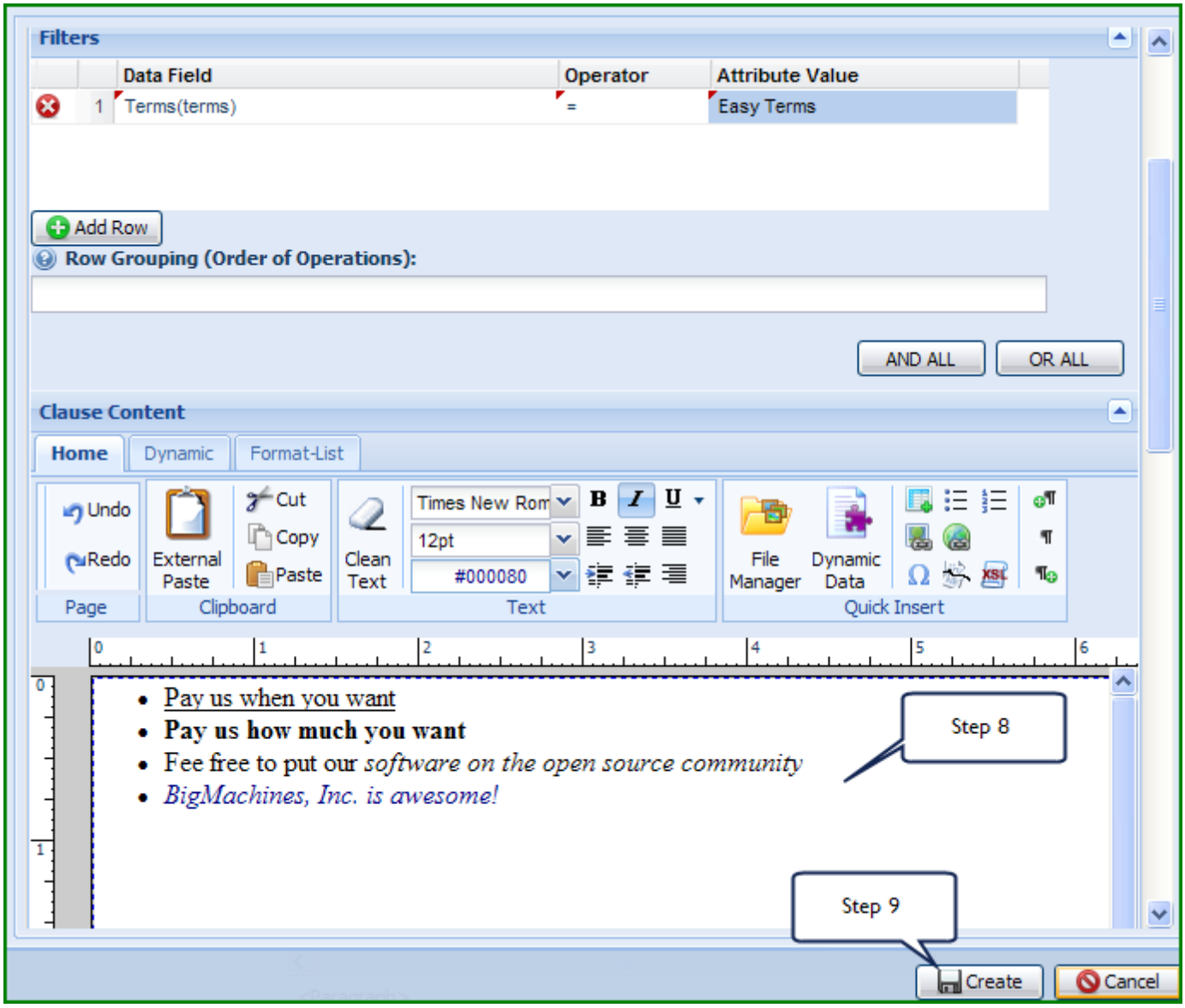
        - You can use parentheses to further control the order of operations.

        - You do not need to use any order for your attribute indexed while defining the order of operations. The attribute indexes can be referenced multiple times in the order of operations to define the correct condition. The order of operation defined is validated by the system when you click Create.

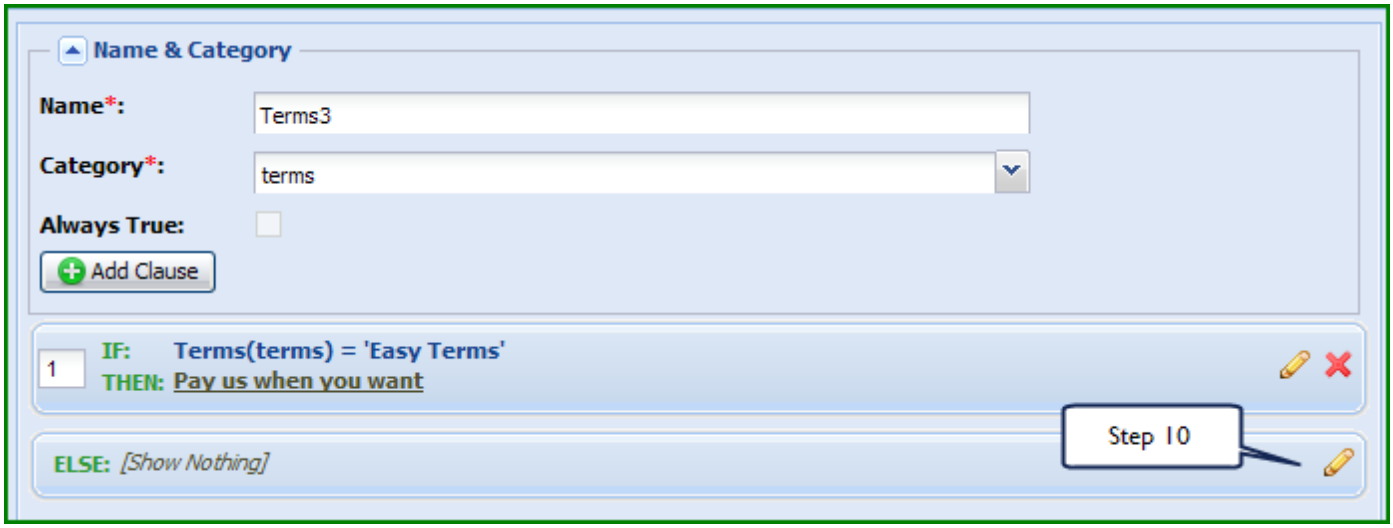


8. Format text, insert hyperlinks, images, tables, dynamic data, repeating sections, and conditionals within the Clause Content section. *This functions similarly to the WYSIWYG editor used throughout Document Engine.* You CANNOT insert a complex conditional.

9. Click **Create**.

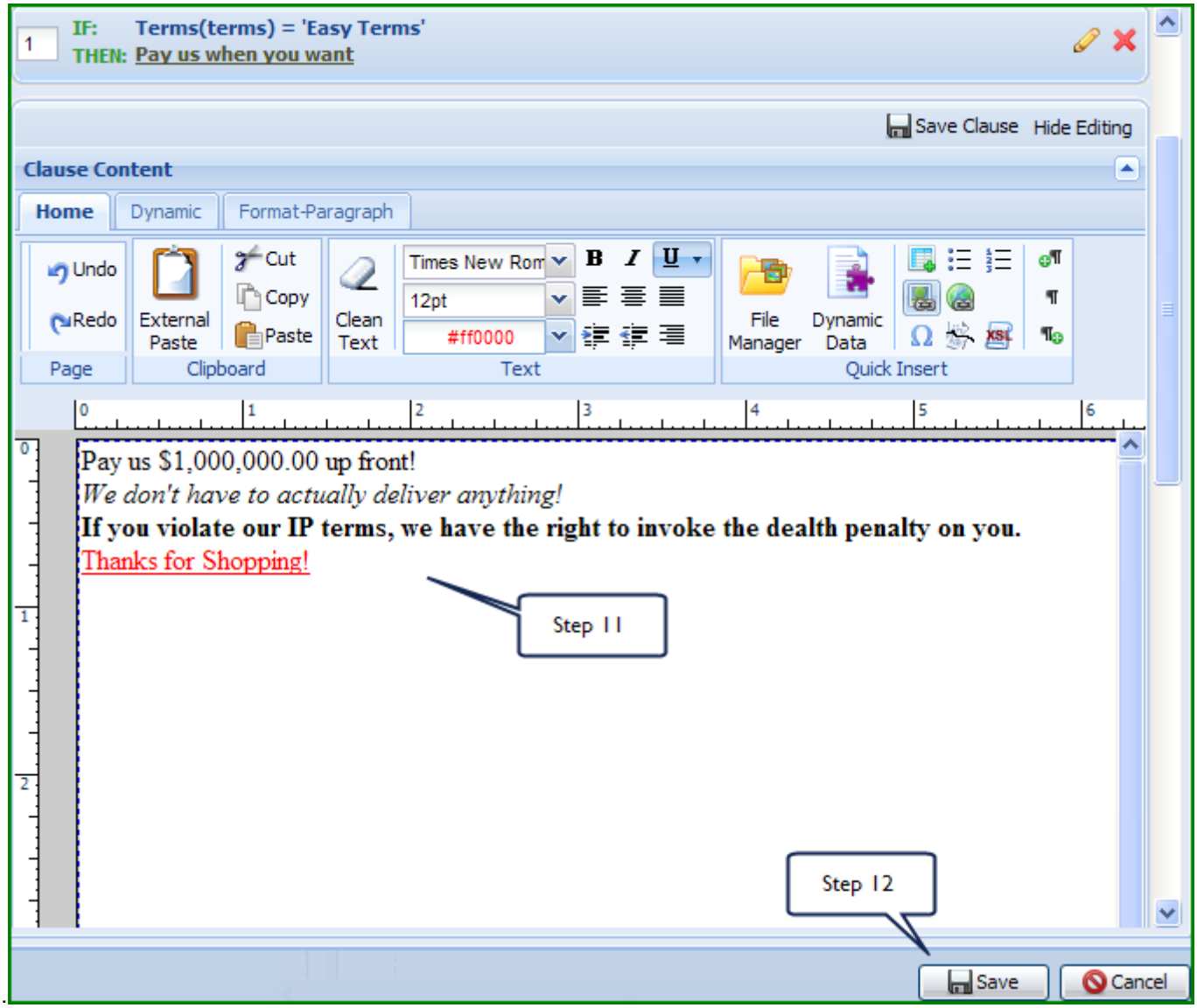


10. Click the pencil icon to add the Clause Content for the "Else" statement.



11. Format text, insert hyperlinks, images, tables, dynamic data, repeating sections, and conditionals within the Clause Content section. *This functions similarly to the WYSIWYG editor used throughout Document Engine.* You CANNOT insert a complex conditional.

12. Click **Save**



Clauses within a complex conditionals can be reordered after creation.

Conditional sections that contain a lot of data or that are nested tables/loops/conditions should be replaced with complex conditionals.

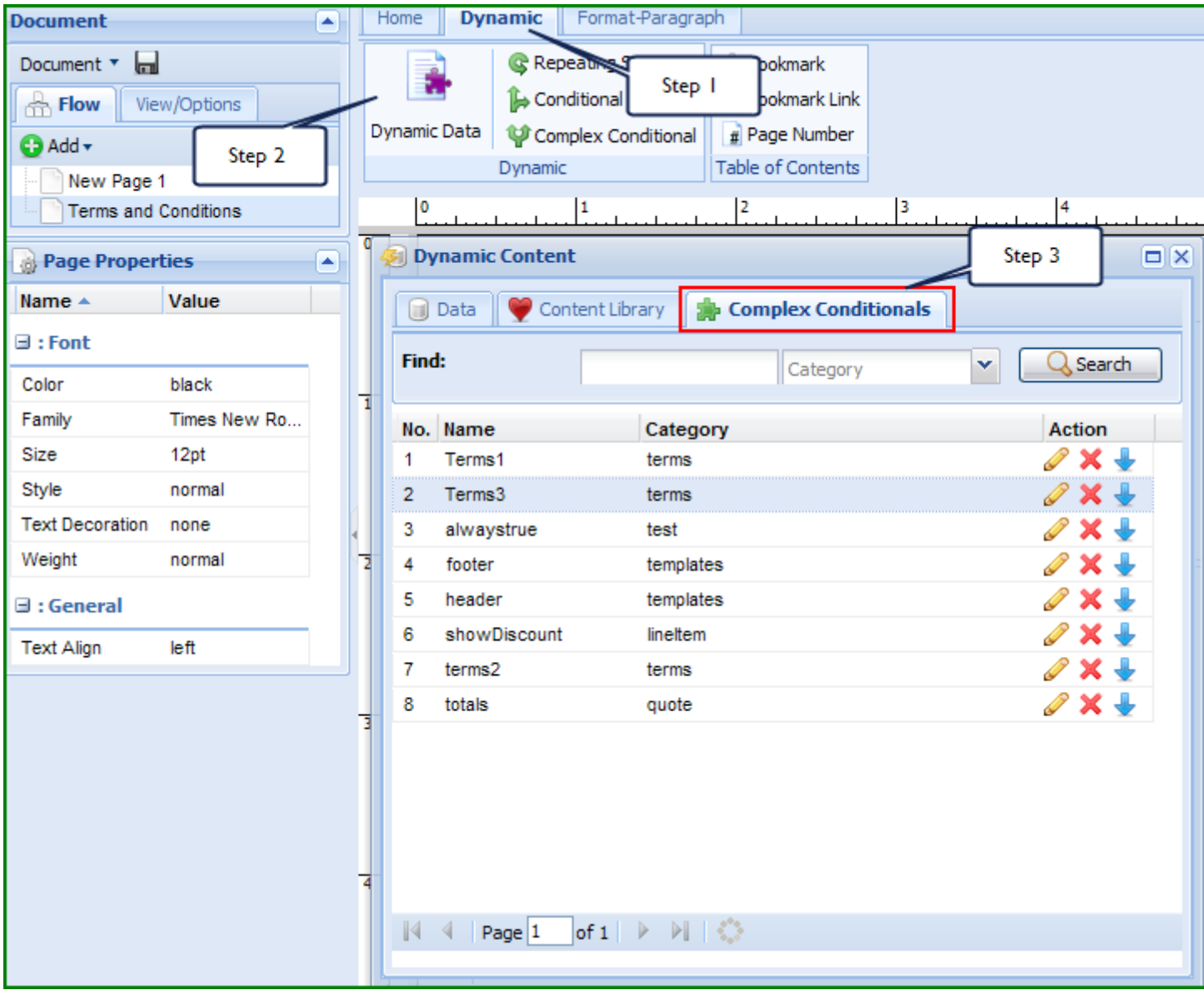
It is Best Practice to store headers, footer, terms and conditions in complex conditionals.

Complex Conditional Library

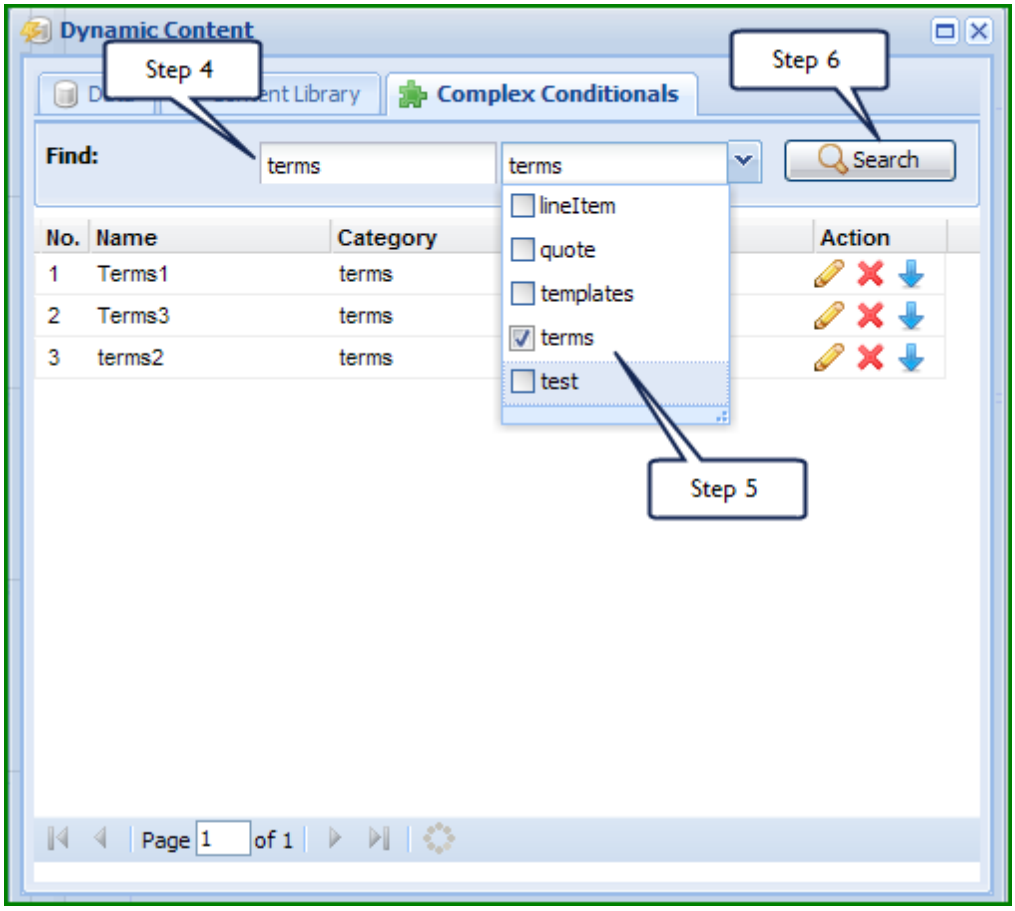
The Complex Conditionals functionality provides the ability to insert complex condition statements with multiple `if...else` clauses inline within the content of a document using the Document Editor without having to resort to custom XSL snippets. Once you have created a complex conditional, it is then stored in the Complex Conditional Library which is found when you navigate to Dynamic Data.

Using the Complex Conditional Library

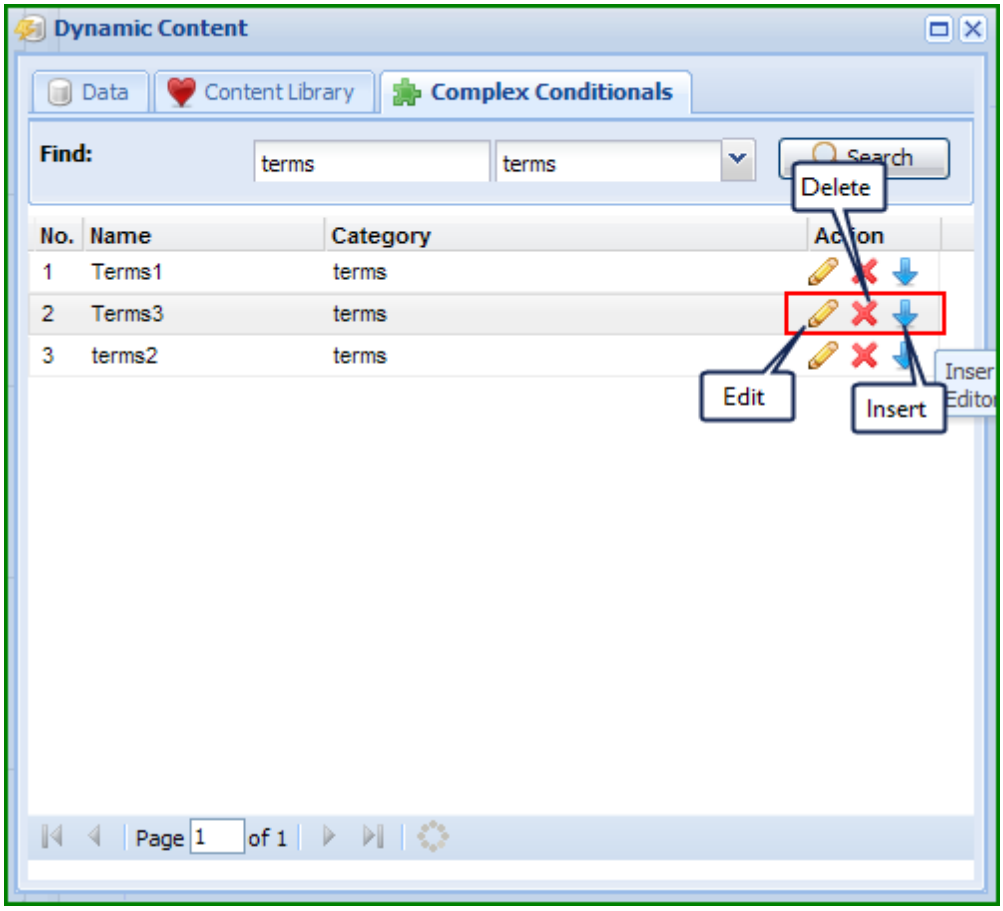
1. From any page in the Document Engine, select the **Dynamic Tab**.
2. Click on the Dynamic Data icon. *You can also access this from the Home Toolbar.*
3. Select the **Complex Conditionals** tab in the Dynamic Content dialog.



4. Search the library for your complex conditional by entering the entire name or part of the name in the text field next to **Find**.
5. Search the library by narrowing down the complex conditionals by category.
6. Click **Search** to list all complex conditionals that meet your search requirements.

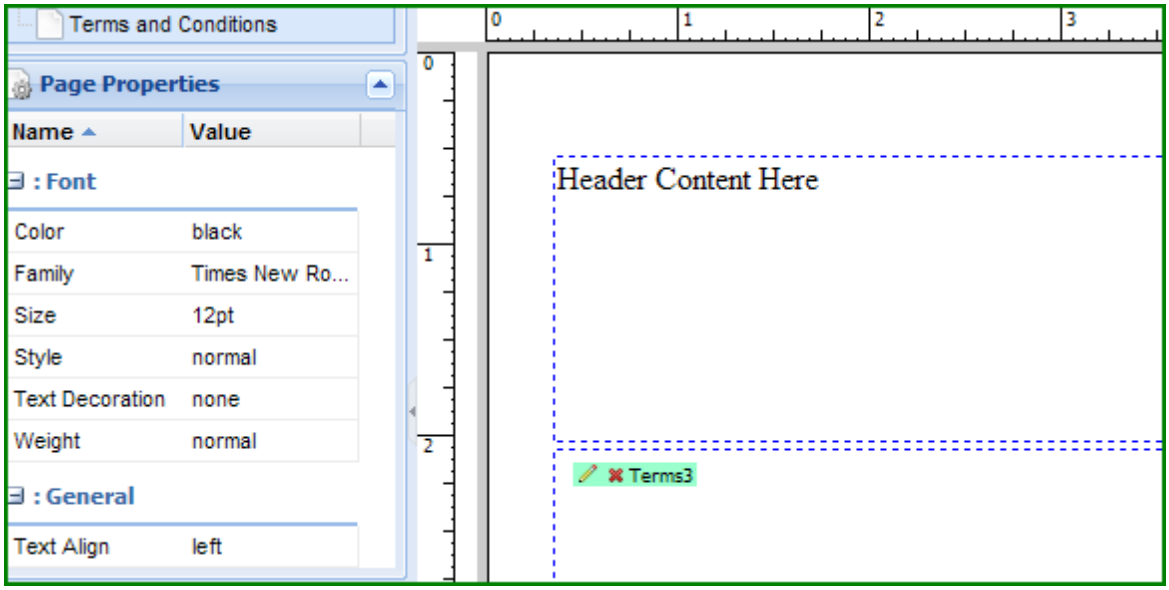


7. Select the complex conditional you search for. *You can either edit, delete or insert this into your page.*



Inserting a Complex Conditional

1. Place your cursor on the page, where you want to insert the complex conditional.
2. Navigate to the Complex Conditionals library and insert a complex conditional into your page by clicking the blue **Insert** arrow.



You will notice that the complex conditional is shown in green. You can edit this by clicking the pencil or delete it from your page by clicking the red "x".

Placing Tables back into Conditionals

If a Table somehow ends up residing outside of the Conditionals they were created in, follow these steps to easily replace Tables within Conditionals. *This allows you to simply recreate the Conditional rather than the entire Table.*

1. Click within the Table that has slipped outside of the Conditional. *The level at which you are editing will appear at the bottom of the Doc Engine.*
2. Click the ”<Table>” button to select the entire table.
3. Select the **Dynamic** tab.
4. Click the **Conditional** button.
5. Enter the conditions under which you'd like the table to appear.

Tables cannot be copied/pasted.

NOTES



Page Number within a Complex Conditional - Page numbers can be dynamically inserted into the content of a complex conditional. The value of the page number within a complex conditional reflect the page number of the current page of the document.

Clauses within a complex conditional can be reordered after creation. Saved Complex Conditionals can be used in all documents within a commerce process.



Content has to be saved once before the user is able to use dynamic data, repeating sections and rows in the Content Editor.

Complex conditionals are shared globally across all documents.

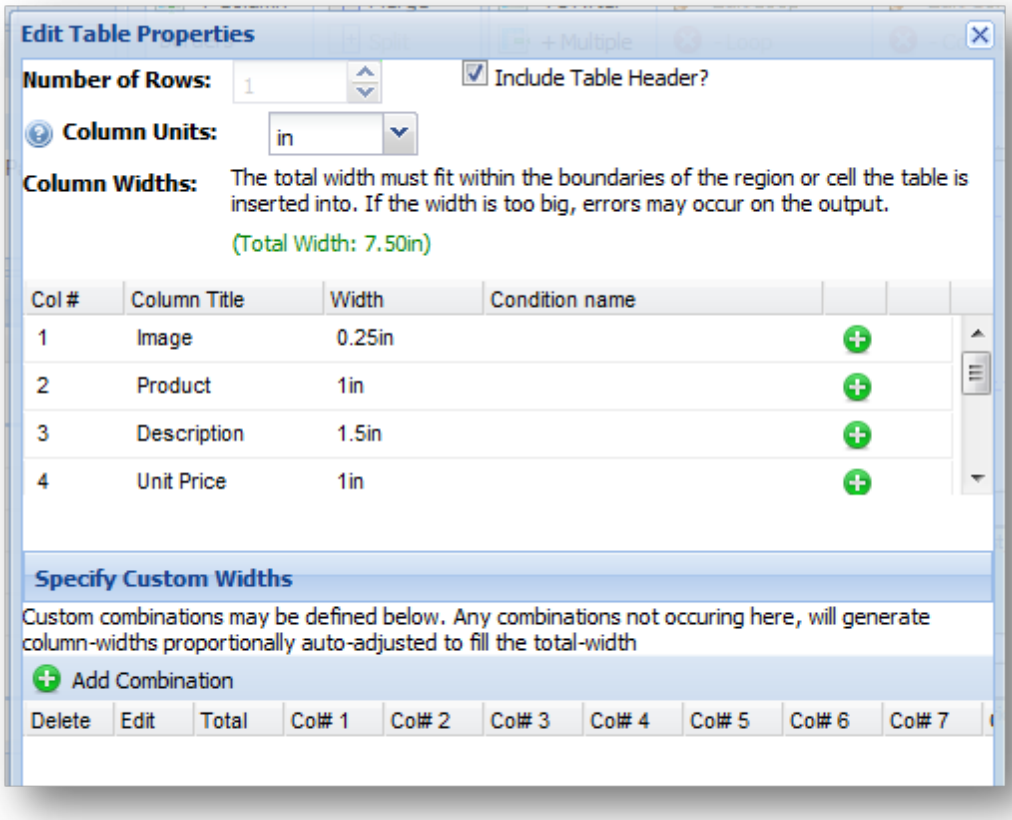
An Always True Complex Conditional cannot be changed to one with multiple conditions after creation - A complex conditional with one or more conditionals can be changed into an Always True type complex conditional. However, a complex conditional that has once been saved as Always True cannot be changed to one with one or more conditional blocks. It is therefore recommended that the fullaccessuser only use the Always True option when certain that no conditionals will be required.

RELATED TOPICS

CONDITIONAL COLUMNS

OVERVIEW

BigMachines allows tables within the Document Engine to contain conditional columns. This allows you to show or hide table columns based on conditions. You also have the ability to specify widths for each column.



ADMINISTRATION

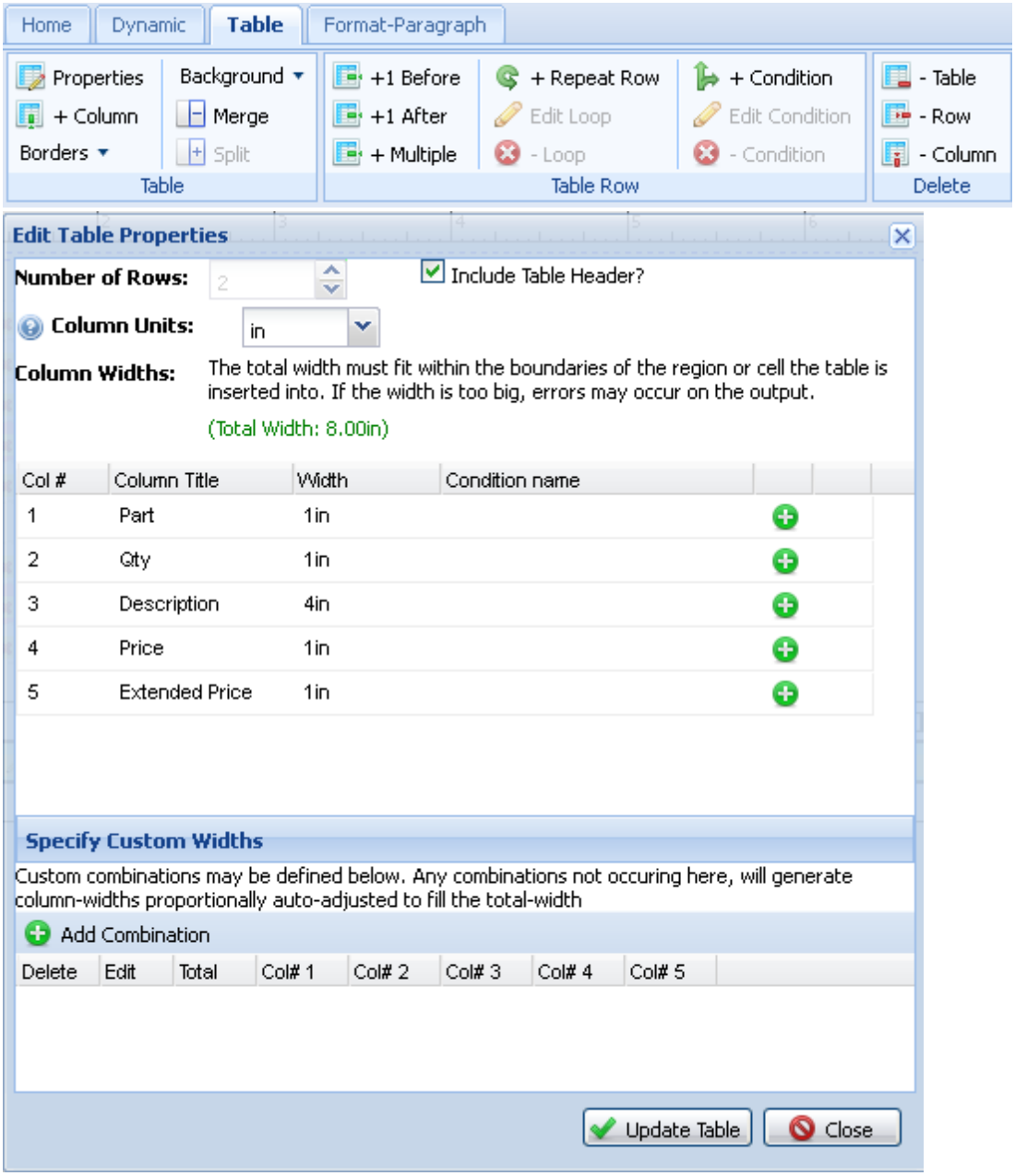
Creating a Conditional Column

You can use conditional columns within a table to control whether you want to show or hide table columns based on conditions. You can also specify custom widths for each column as well as create combinations. This is done by editing the table properties. In order to better explain how conditional columns work, we'll start with a table that has already been created.

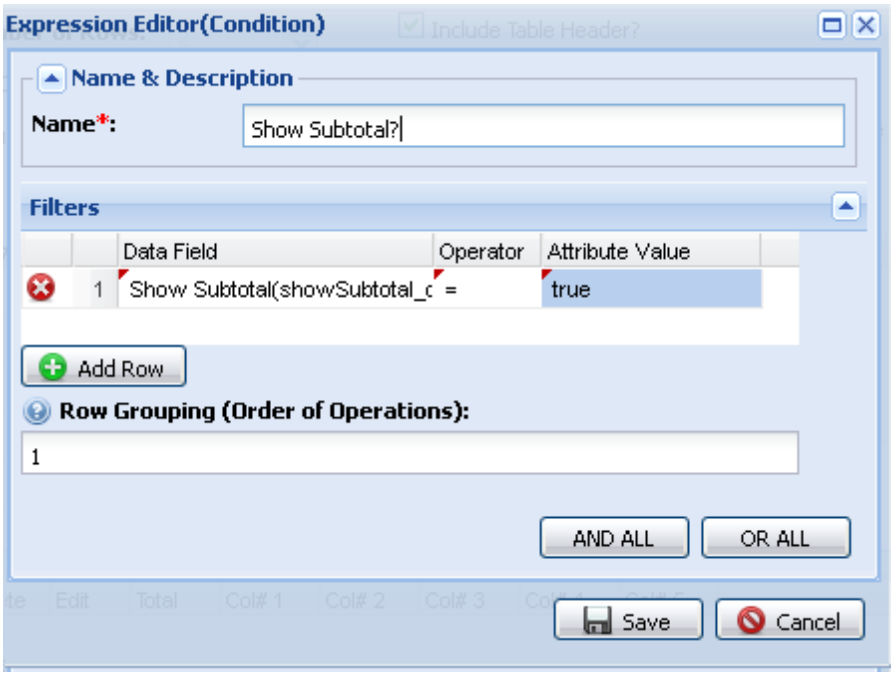
	/bill_address_2
	/bill_city , /bill_state_province
Ship To:	/ship_address_1
	/ship_address_2
	/ship_city , /ship_state_province /ship_zip

Part	Qty	Description	Price	Extended Price
/_part_number /_model_name	/_price_quantity	/_part_desc	/listPrice_line	/extendedNetPrice_line

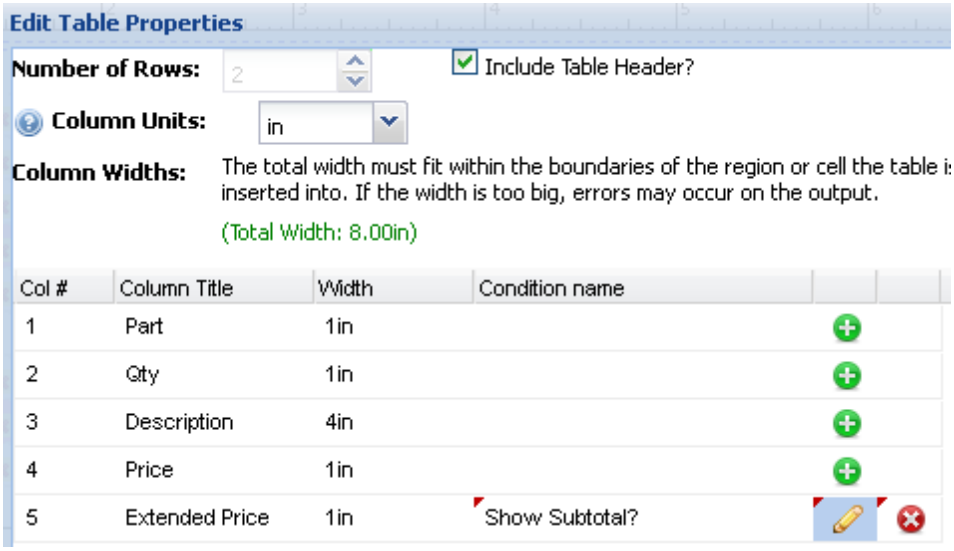
1. After you have created a table, select the **Table** tab and select **Properties**. *This will open up the Edit Table Properties dialog.*



2. Select a column on which to write a condition. *In this case, we're going to use Extended Price.*
3. Click the Green Plus icon next to the column to open the Expression Editor dialog.
4. Enter a **Name** for this condition.
5. Enter one or more filters on which the condition is based. *In this example, our Data Field will be Show Subtotal(showSubtotal_quote)=true.*



6. Click **Add Row** to add additional filters for the condition (if necessary).
7. Enter Row Grouping (Order of Operations). *The default for one row is 1.*
8. Click **Save** to save the condition and close the window.

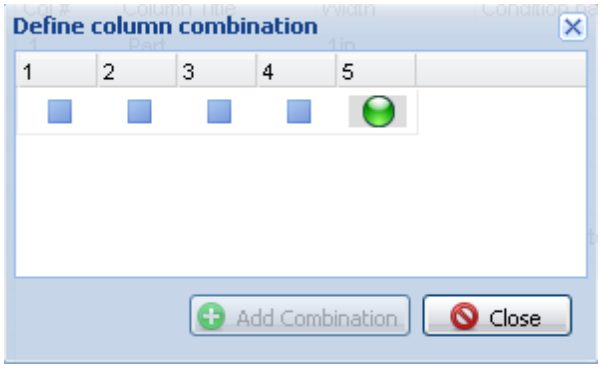


Notice that the Condition Name column now has an entry and the Green Plus icon is now a pencil. To edit the condition, click the Pencil. To delete the condition, click the Red X.

Creating a Combination

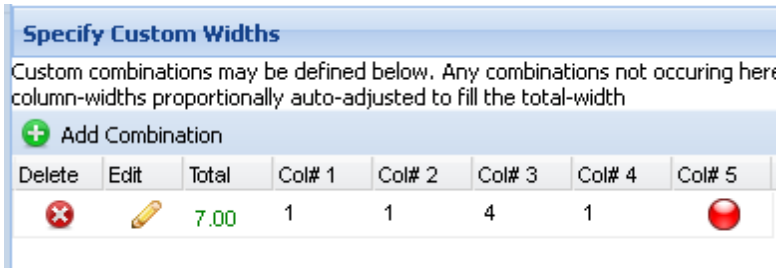
The Edit Table Properties window provides a visual UI to create the combinations, so you can see which columns will appear where and when. It is possible to create multiple combinations.

1. After you have created one or more conditional columns, click **Add Combination** under Specify Custom Widths. A dialog titled Define column combination will open. *Notice that in column 5, you'll see a Green circle. This means, that whether or not the condition is met, column 5 will be displayed.*



2. Click the Green circle if you'd like to hide the column from the table when the condition is not met. *It will turn red to indicate the column will be hidden from view.*

3. Click **Add Combination**. The dialog will close and your new combination will be visible on the Edit Table Properties window.



4. Edit the column widths for the columns that will be displayed so that they equal the total width of the table. *In this case, the total table width is 8.00in. The total that is displayed in the image above is 7.00, indicating that one or more columns can be adjusted to make up for the width of the hidden column 5.*

5. Click **Update Table** to save and close your changes.



Once you have created at least one conditional column, you can define custom combinations.

Combinations without a specific width set will have the columns auto-adjusted.This allows you to remove the column if the condition is not met. If the standard table size is larger than the page, and no combinations are specified, the table will be larger than the page.

NOTES



It is recommended that you click "update" on your table pop-up after making any updated to conditionals to avoid any data corruption issues.



There is a maximum of 30 columns, with a condition, per table.

Merged columns cannot have a condition.



If the standard table size is larger than the page and no combinations are specified, the table will run off the page.

Conditions must be deleted before the column can be deleted.

There is no indication within the UI if a table column has a condition defined on it. You'll have to open the Table Properties to view all conditions.

RELATED TOPICS

BEFORE BEGINNING BML

OVERVIEW

There are some things to note before attempting to use BML.

Shortcuts

Select the question mark in the toolbar above the **Script Definition Area**, to see the following menu of commands:

SHORTCUT	What does it do?
Tab	Add tabulation to text
Shift + Tab	Remove tabulation from text
CTRL + F	Search next/open search area
CTRL + R	Replace/Open search area
CTRL + H	Toggle syntax highlight on/off
CTRL + G	Go to line
CTRL + Z	Undo
CTRL + Y	Redo
CTRL + E	About
CTRL + Q	Close pop-up
Access Key E	Toggle editor <

Standard Syntax

You will see some standard syntax throughout the function editor:

```
function(param 1, param 2, param 3, [optional param])
```

Example: `substring(str, start, [end])`

Parameter	Data Type	Definition
str	String	This is the given string for...
start	Integer	This represents the index...
[end]	Integer	This represents the index...

Common Operators

Operator	Looks Like
equals	== 12 == 12

	"string" == "string" true == true
not equal	<> 12 <> 13 "string" <> "abc" true <> false
less than	< 12 < 13 "13" < "14"
greater than	> 12 > 13 "13" > "14"
less than or equal to	<= number <= number string <= string
greater than or equal to	=> number => number string => string
and not or	<boolean> and <boolean> <boolean> not <boolean> <boolean> or <boolean>

Numeric Operators

Numeric operators always return a numeric data type. Numeric inputs include: numeric literals, numeric identifiers, and numeric functions. Numeric operators can also be used whether a numeric value can be plugged in or not, such as: in other functions, relational operators, and expressions. For more information, see Numeric Functions.

Numeric Operator	Meaning	Looks Like
+	Addition	12 + 12
-	Subtraction	15 - 12
*	Multiplication	15 * 12
/	Division	15 / 12
%	Modulus	<number> % <number>

Example:

```
sqrt(45 + 45)
(3 * 3) + (4 * 4)
(3 + 4) == (4 + 3)
```

String Operators

There is one string operator that can be used to return a string as an output. The string operator for concatenation is **+** and its formation is:

```
<string> + <string>
```

Literals

Literals can be added to scripts when following these conventions:

Literal	Data Type	Examples
Integer	Numeric	12345
Floating point number	Numeric	123.45 12345 .345
String	String	"volume" "23 psi" "100"
True	Boolean	True
False	Boolean	False

RELATED TOPICS

WHAT IS BML?

OVERVIEW

BML (BigMachines Extensible Language) is a scripting tool that is used to capture a company's complex business logic within BigMachines Configuration and Commerce. BML is based of many different programming languages. You will see syntax that is familiar in Java, Basic, Python, and C++, but they aren't exactly the same. After reading the section on BML, you should have a good understanding of what programming language is and how it's used and structured. You should also understand the syntax of BML and have generated some very basic and functional code.

What is a function?

A function is usually defined as a rule that relates one variable value to another. In programming, it also represents a series of commands that perform an action or calculate a value depending upon certain parameters. Parameters are the values that the function uses. But, not all functions require parameters. For more information, see Function Editor Basics.

What is a variable?

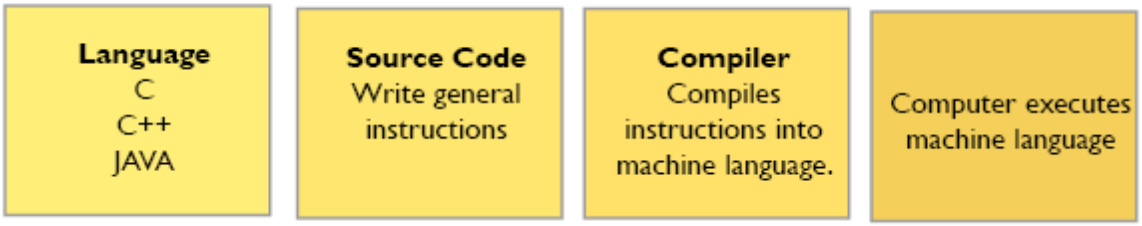
Variables are placeholders for the data, which a program may use or manipulate. They are given unique names so that they can be assigned values and referred to throughout the source code. In order to use a variable, you must declare it by giving it a unique name. In many programming languages, once a variable is declared, it must be initialized. However, in BML, these two steps occur simultaneously.

ADMINISTRATION

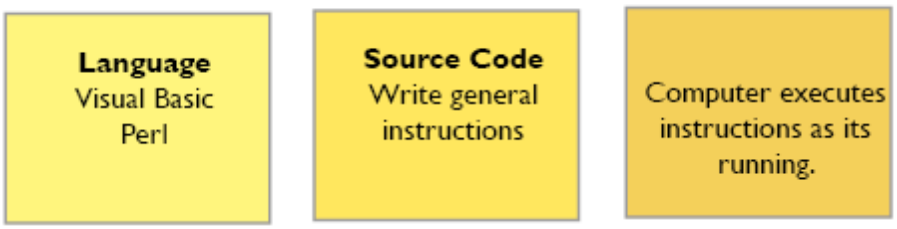
Programming Language

A programming language is an artificial language that is used to write instructions that a computer can understand. Programming languages can be broken down into two groups:

1. **Compiled programming:** this is a language that must be converted into machine language, after it's been written by the programmer, so that a computer can understand and execute the instructions. These languages require the use of special software (called a compiler) prior to being executed.



2. **Interpretive programming:** this language do not have to be compiled. The computer understands and executes the instructions while the program is running. See examples below.



BML has characteristics of both languages. It's a Java based mark-up language, so it does need to be compiled, but that is done within the application.

Programming Structure

BML follows the Structured computer programming model (not Object-Oriented model). Structured programming uses blocks of code that are executed one after the other. They include control statements which dictate the flow, or when these statements are executed. Almost all structured programming, including BML, shares a similar overall structure. They include:

- Statements to establish the start of the program
- Variable declarations
- Source code (blocks of programming statements)
- In BML (a return statement is always required)

Basic Data Types

Almost all programming languages include the use of data types. A data type determines the values that a variable can contain and how it is executed. A data type also represents a constraint placed on the interpretation of data. There are four basic data types:

- **Integer:** Data is stored as a whole number. When assigning any value that contains a decimal or a different character, it's either rounded or an error is thrown.
- **Float:** Data is stored and/or returned with a decimal.
- **String:** Allows text output of any kind to be returned. In BML, string variables will always reside within double quotations ("").
- **Boolean:** Returns either a True or False, based on your inputs.

Common Statement Elements

Element	Description
;	Represents and end statement. This is required on each line of code.

Element	Description
//	Indicates a comment line.
return	This is the beginning of the return statement which is required at the end of your code. BML will not process instructions without it.
print	This function displays all data types in the console to make debugging easier. You can give it direct inputs or have it print variables and results from other functions. It displays as purple in the Debugger

You may also see the element `return "";`. This will return an empty String if there isn't anything to return. It will only return a String (signified by double quotation marks), not an Integer, Float, Boolean, or Array.

Basic BML Syntax

The basic structure includes statements that comprise the source code and ends with a return statement. The example below shows variables, print statements, and a return statement.

```
1  var_1 = 124.5;
2  var_2 = false;
3  var_3 = 5;
4
5  results = "This is a float to string " + string(var_1);
6          print results;
7
8  results_2 = "This is a boolean to a string " + string(var_2);
9          print results_2;
10
11 results_3 = "This is an integer to a string " + string(var_3);
12         print results_3;
13 return "";
```



It is considered a Best Practice to add spaces and comments to your code. This is done to make the code more readable.

RELATED TOPICS

BEFORE BEGINNING BML

OVERVIEW

There are some things to note before attempting to use BML.

Shortcuts

Select the question mark in the toolbar above the **Script Definition Area**, to see the following menu of commands:

SHORTCUT	What does it do?
Tab	Add tabulation to text
Shift + Tab	Remove tabulation from text
CTRL + F	Search next/open search area
CTRL + R	Replace/Open search area
CTRL + H	Toggle syntax highlight on/off
CTRL + G	Go to line
CTRL + Z	Undo
CTRL + Y	Redo
CTRL + E	About
CTRL + Q	Close pop-up
Access Key E	Toggle editor <

Standard Syntax

You will see some standard syntax throughout the function editor:

```
function(param 1, param 2, param 3, [optional param])
```

Example: `substring(str, start, [end])`

Parameter	Data Type	Definition
str	String	This is the given string for...
start	Integer	This represents the index...
[end]	Integer	This represents the index...

Common Operators

Operator	Looks Like
equals	== 12 == 12

	"string" == "string" true == true
not equal	<> 12 <> 13 "string" <> "abc" true <> false
less than	< 12 < 13 "13" < "14"
greater than	> 12 > 13 "13" > "14"
less than or equal to	<= number <= number string <= string
greater than or equal to	=> number => number string => string
and not or	<boolean> and <boolean> <boolean> not <boolean> <boolean> or <boolean>

Numeric Operators

Numeric operators always return a numeric data type. Numeric inputs include: numeric literals, numeric identifiers, and numeric functions. Numeric operators can also be used whether a numeric value can be plugged in or not, such as: in other functions, relational operators, and expressions. For more information, see Numeric Functions.

Numeric Operator	Meaning	Looks Like
+	Addition	12 + 12
-	Subtraction	15 - 12
*	Multiplication	15 * 12
/	Division	15 / 12
%	Modulus	<number> % <number>

Example:

```
sqrt(45 + 45)
(3 * 3) + (4 * 4)
(3 + 4) == (4 + 3)
```

String Operators

There is one string operator that can be used to return a string as an output. The string operator for concatenation is **+** and its formation is:

```
<string> + <string>
```

Literals

Literals can be added to scripts when following these conventions:

Literal	Data Type	Examples
Integer	Numeric	12345
Floating point number	Numeric	123.45 12345 .345
String	String	"volume" "23 psi" "100"
True	Boolean	True
False	Boolean	False

RELATED TOPICS

LIBRARY FUNCTIONS

OVERVIEW

The BML Function Library enables the user to write efficient and reusable custom BML functions. The user can write and store BML functions in a central library and call these functions from different areas in BigMachines.

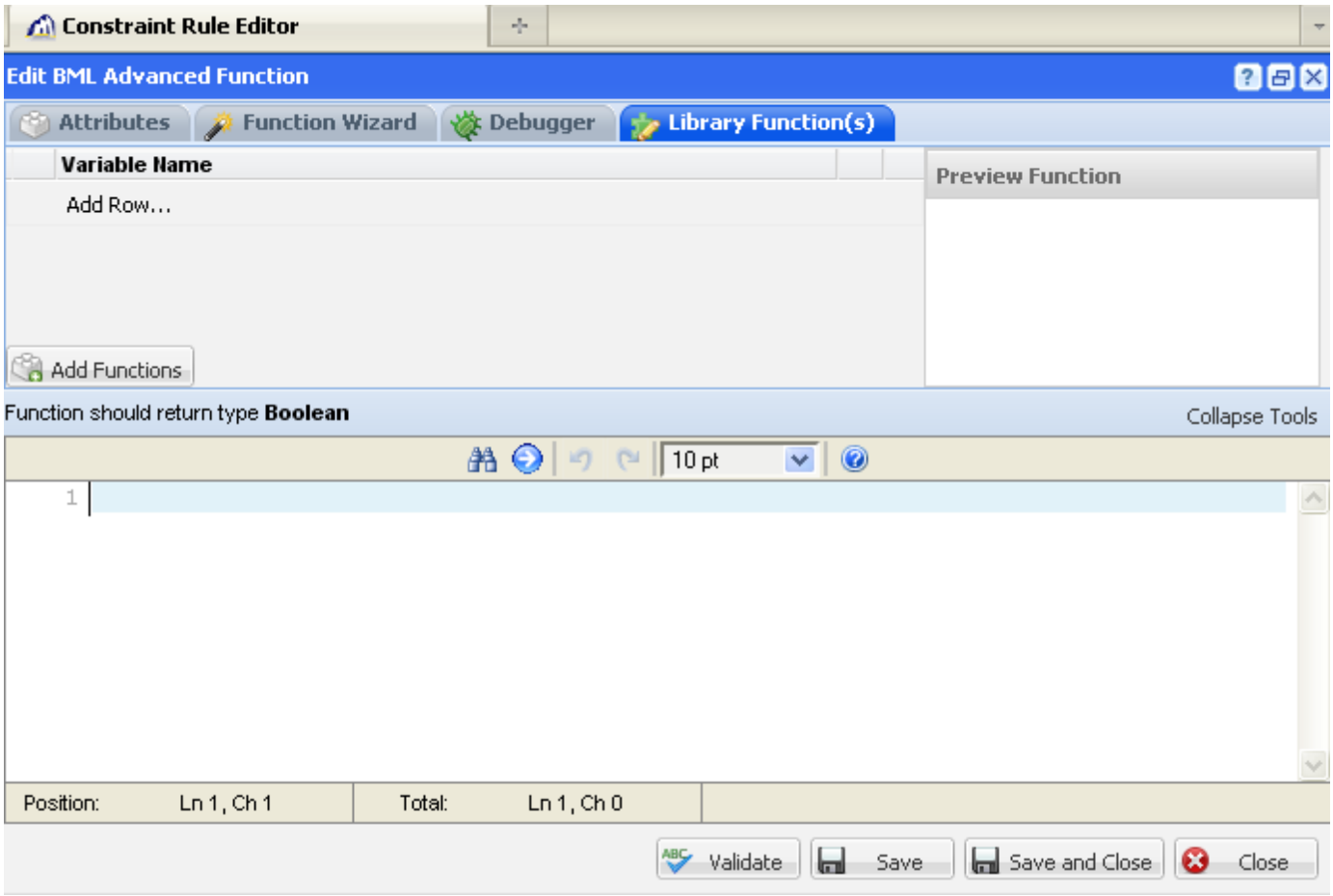
- **Util BML Library** - The functions in this library can be accessed and called throughout BigMachines.
- **Commerce BML Library** - The functions in this library are accessible only to a main document-sub document pair. Each Commerce BML library has a 1:1 relationship with a main document and sub-document pair. There can be as many Commerce BML libraries as there are document pairs in Commerce. *The Commerce BML Library is not available until a main document has been associated with a sub-document.*

ADMINISTRATION

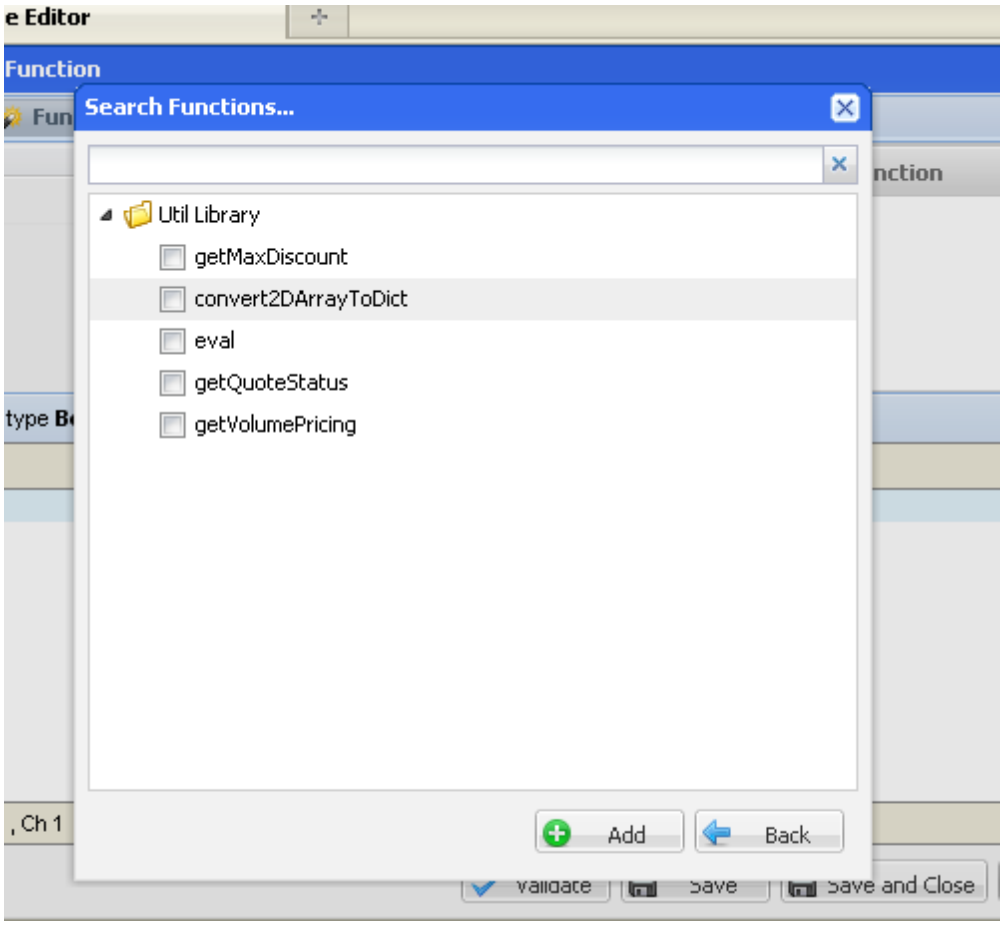
Using Library Functions in Configuration BML

Once a Util Library Function has been defined, it's available for use in the **Library Function(s)** tab within the Function Editor.

1. Click **Add Functions**to choose from a list of Util Functions. *A pop-up dialog will open.*



2. Click one or multiple Util Functions.



3. Return to the Util functions area and make sure you've deployed your function(s) if desired function does not appear.

Using Library Functions in Commerce BML

When you click **Define Function** in Commerce, a window will appear asking you to select attributes. You can select **System Variable Name**, **Variable Name for (Quote)**, **Varible Name for (Line)**, and **Library Function(s)**.



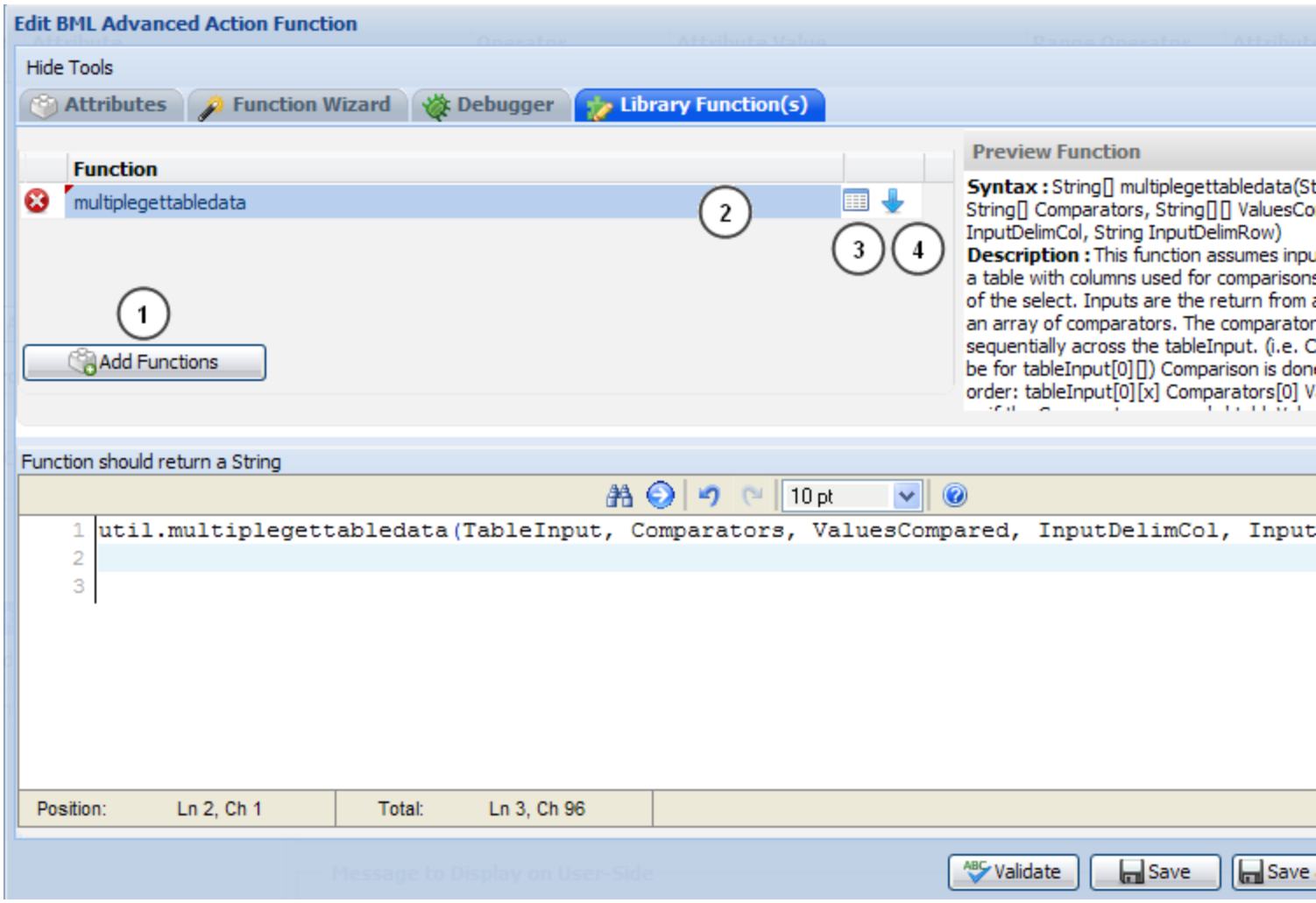


You can add Util and Commerce Library Functions by clicking the **Re-Select** button within the BML Function Editor.

Adding a Library Function

1. Click **Add Function**.
2. Choose desired function from drop-down menu.
3. Select the square icon to preview the function in the pane to the right.

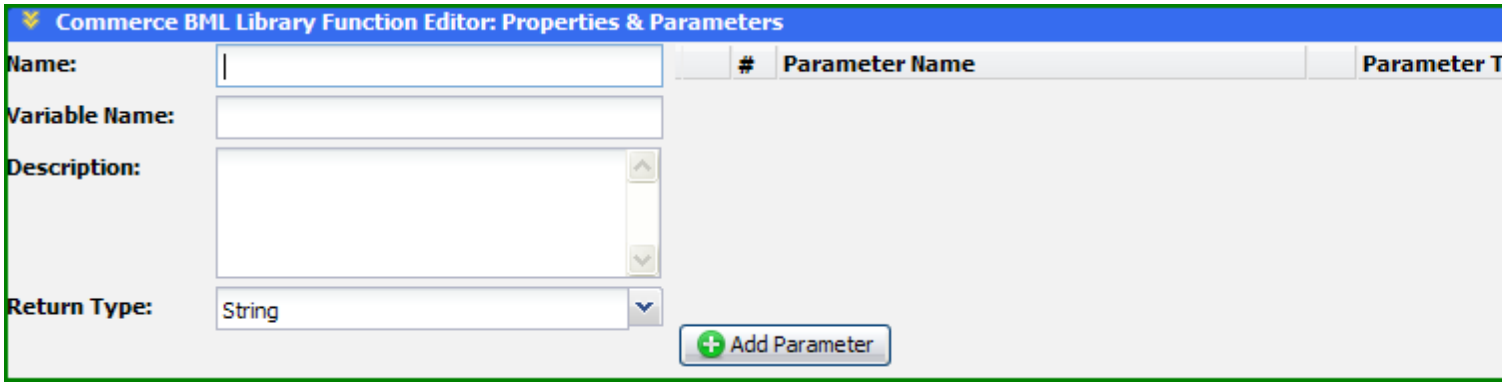
4. Select the blue arrow to insert the function into the **Script Definition Area**.



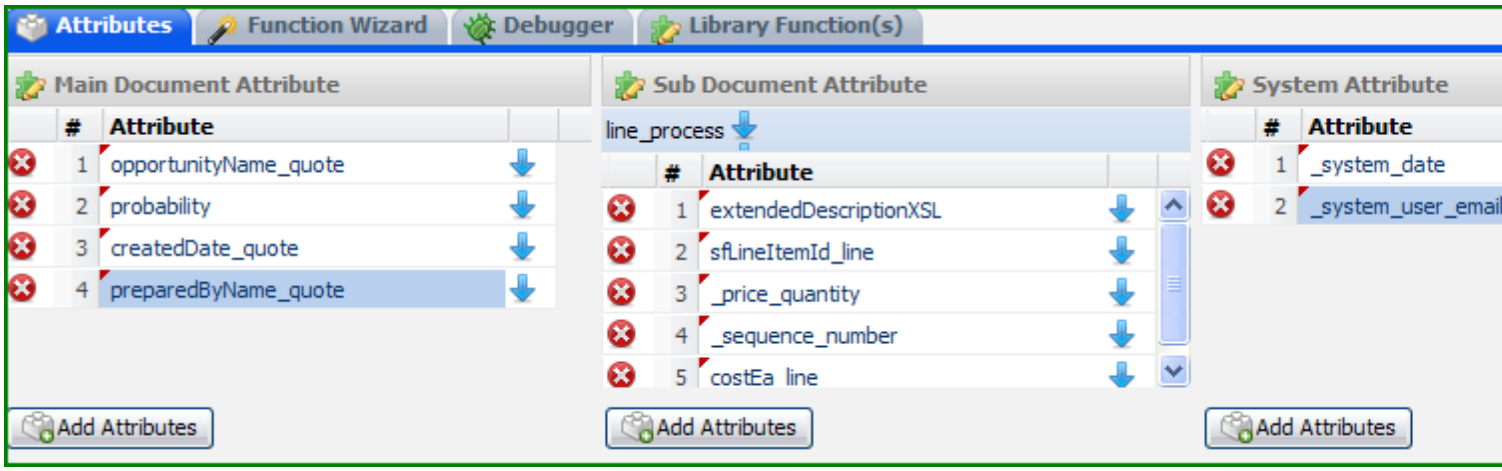
Adding a Commerce Library F(x)



1. Navigate from: Process Definition > Process > Document > Library Functions > List.
2. Select **Add** to create a new commerce library function.
3. Create a **Name**, **Variable Name**, **Description** and choose a **Return Type**.
4. Select**Add** in the **Parameter(s)** section if the function requires parameters. *Pick a parameter name and choose a data type.*
5. Reference the parameter in the library function with the name that you specify here.



6. Select **Add Attributes** to access the main document attributes in the **Main Document Attribute** section.



7. Select **Add Attributes** to access the sub-document and the system attributes in the **Sub-Document Attribute** and **System Attributes** sections.

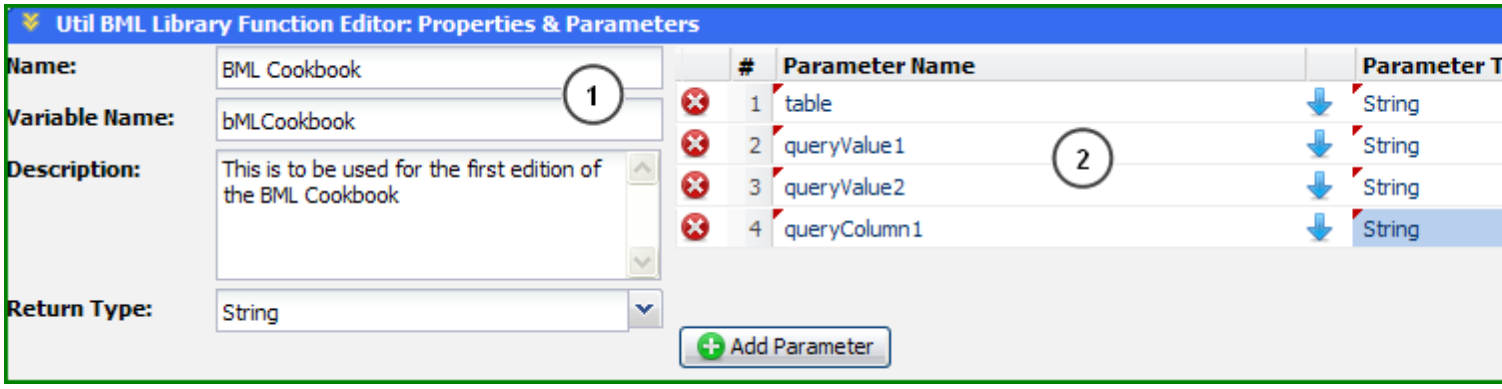


The **Main Document**, **Sub-Document** and **System Attributes** sections are only available in the **Commerce Library Editor**. They are not available in the **Util Library Function**.

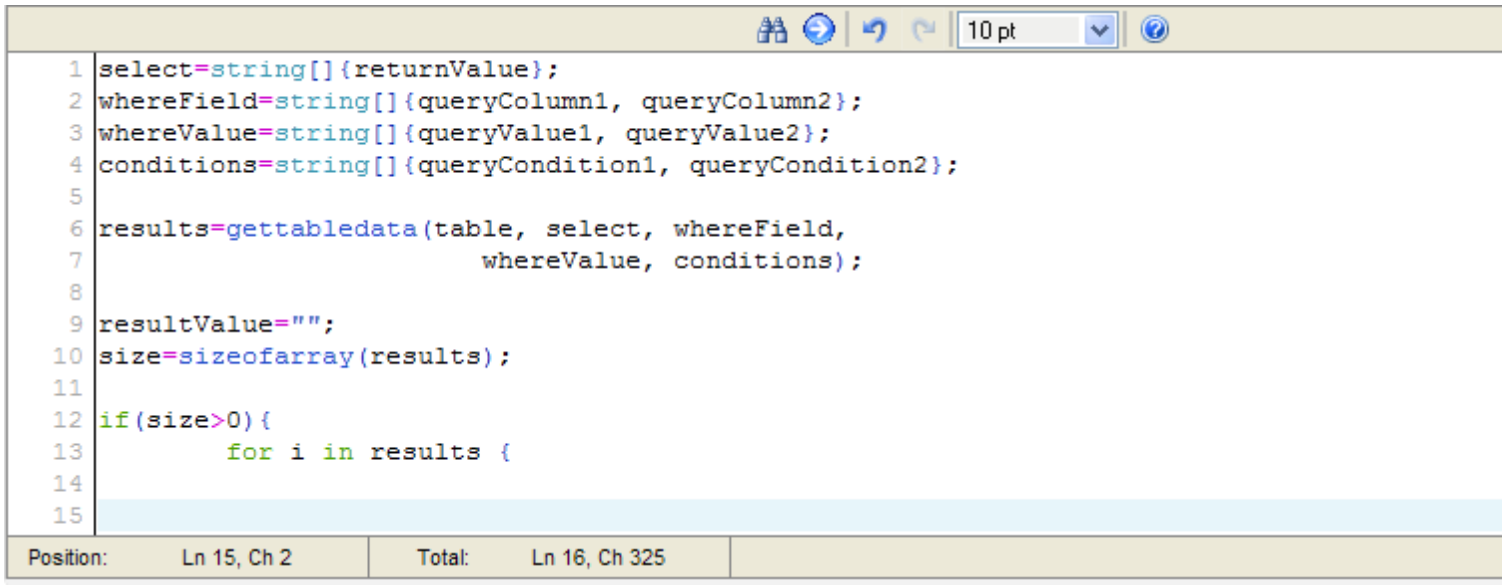
Adding a Util Library F(x)

Navigation: Admin Home Page > Data Tables & Global Functions > BML Library

1. Create a **Name**, **Variable Name**, **Description** and choose a **Return Type**.
2. Add the necessary parameters.



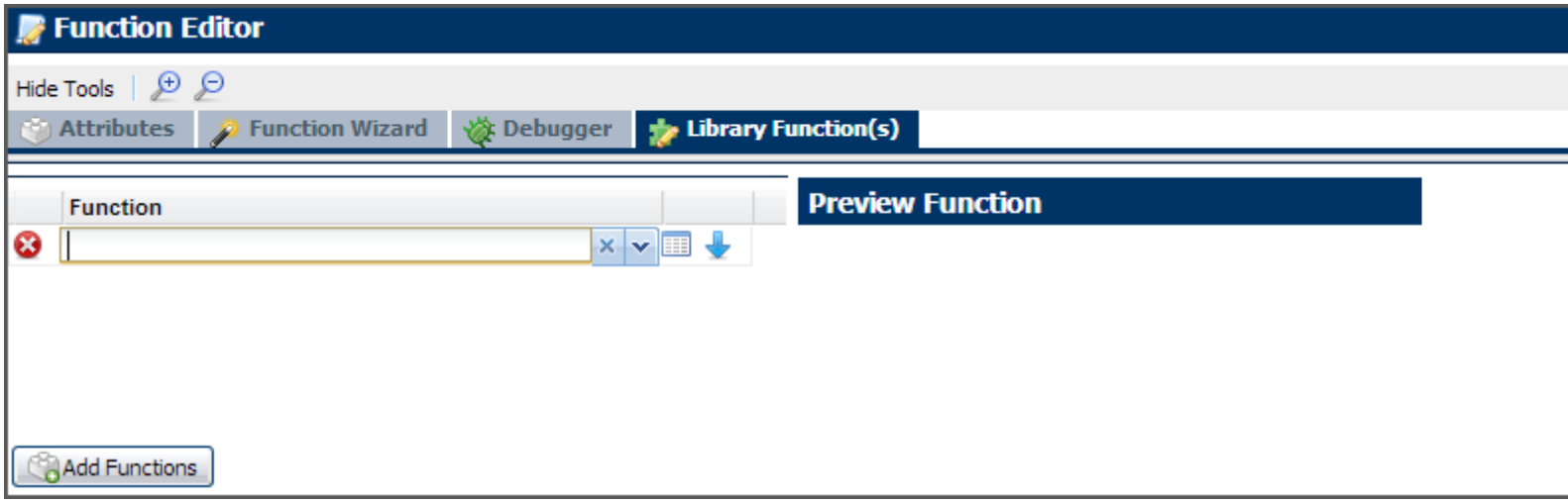
3. Create your script, adding attributes as necessary.



4. Select **Add** to make your new library function available for use.

Function to Function Calls

Util and Commerce Library Function Editors utilize Function to Function calls. Function to Function calls allow administrators to compartmentalize BML when dealing with complicated configuration or quoting scenarios. This feature will assist with the organization of BML and provides a solution to the compiled Java class size-limit issue. *Function to function calls mimic the behavior of a Modify function calling a Library function.*



Adding a Function to Function Call

1. Open the **Function** drop-down menu.
2. Select a Library Function to **Insert into BML**. The options are:
   - **Commerce Library functions:** Only available in Commerce Library Function Editors. Commerce Library functions may only be called by Commerce Library functions.
   - **Util Library functions:** Available in both Commerce and Util Library Function Editors.
3. Once the function is viewed using the **Preview Function**, click **Insert into BML**blue arrow button.
4. When finished, click one of the following buttons:



The Function to Function link appears on the Related Rules page. *When it is referenced by other Utils it will be displayed on the Related Rules page.*

Recursive validation is performed during the following for Util and Commerce Library Functions: Validate, adding, applying, or updating a function.



Util and Commerce Library functions cannot self-reference. *Recursive calling of the same Util and Commerce Library functions will fail and result in a compilation error when called at any point in the reference chain. Util and Commerce Library functions will not appear in the Import list for themselves.*

NOTES



Commerce Library functions can call other Commerce Library functions. *Commerce Library functions can call Util Library functions.*

Util Library functions can call other Util Library functions. *Util Library functions cannot call Commerce Library functions.*



Library functions can be created if there isn't a pre-defined function for what you are trying to accomplish.



Library functions must be created before they can be added.

RELATED TOPICS

DEBUGGING UTIL FUNCTIONS

OVERVIEW

BML libraries are extremely important in that they allow us to avoid duplicate code, and to easily maintain discrete logical operations. They are different than Advanced Modify functions and Advance Validation rules, and it may take some extra effort to reproduce errors.

BML libraries can take Arrays and Dictionaries as parameters. There is no UI interface for representing these objects, so you will have to create a test script to call the function.

ADMINISTRATION

Before you begin

The process to debug a BML library will generally be the same:

1. Find the original function call that failed.
2. Identify what parameters were passed in.
3. Recreate the call in the test script and run the debugger.

Be aware in your testing that objects are passed by reference. This means that arrays and dictionaries that are modified in a library will be changed in the calling context. **For example:**

```
//in util library "passByRef"
// param myDict {dict("string")} An object to be modified;
put(myDict, "key2", "val2");
return "This is the return string.";
// in advanced modify function
myDict = dict("string");
put(myDict, "key1", "val1");
res = util.passByRef(myDict);
print res; // will be "This is the return string."
print myDict; // will be "{key1=val1, key2=val2}"
```

Example

Assume that when a user clicks Save they get an error "Divide by 0". The error logs identify the problem as occurring in the Save action.

1. Track down Save's Advanced modify and see the error occuring when you call util.getAverage:

```
pricesArray = util.parsePricesDelimitedString(pricesDelimitedString_quote, pricesDelimiter_quote);
averagePrice = util.getAverage(pricesArray); // error occurs here
//[...] rest of save function
```

2. Get a copy of the parameters being passed in. You can do this by printing the array:

```
pricesArray = util.parsePricesDelimitedString(pricesDelimitedString_quote, pricesDelimiter_quote);
print pricesArray; // will show up in the debugger
averagePrice = util.getAverage(pricesArray); // error occurs here
//[...] rest of save function
```

3. Let's assume that we print out the array, and get back the value "[]". We are now ready to go debug the util library directly. Say the util library looks like so:

```
/*
* Util library 'getAverage'
* @param numbers {Integer[]} Array of numbers to be averaged
* @returns {Integer} Average of all the numbers in the array
*/
count = sizeofarray(numbers);
total = 0;
for num in numbers {
total = total + num;
}
result = total/count;
return result;
```

4. We can't run the debugger directly, because of the array parameter. So we will write a test script instead. Some notes about writing test scripts:

5. Write a test that passes or fails. Do this by specifying an expected value, and comparing it to the result of the function.

6. Leave other test scripts in place; you can use tests written previously to help identify conflicts in your code.

```
// test script for util lib getAverage
//create a blank array per the results in the Save action
myArr = integer[];
//when we pass in a blank array, we expect to get 0 back
actual = util.getAverage(myArr);
```

```
//test can pass or fail - print a meaningful message if there is a problem
expected = 0;
if(expected != actual) {
print "Test Failed: expected value 0 for empty array, got: " + actual;
} else {
print "Test Passed: empty array returns value of 0.";
}
return res;
```

Now when we run our test script, we get the expected "Divide by 0" error. We've successfully reproduced the error, and have also used the test script to describe the desired behavior in a meaningful way. We can now fix the error, which occurs when "count" is equal to zero:

```
/*
* Util library 'getAverage'
* @param numbers {Integer[]} Array of numbers to be averaged
* @returns {Integer} Average of all the numbers in the array
*/
count = sizeofarray(numbers);
total = 0;
//Empty arrays should return a value of 0
if(count == 0) {
return 0;
}
for num in numbers {
total = total + num;
}
result = total/count;
return result;
```

USE CASE

Util Library Debugger

1. Obtain required information for the Util Function.

    a. Locate the function that contains the Util Function you wish to debug. *This could be a Commerce function, such as Pricing, or a function on an action or attribute.*

    b. Locate the function that contains the Util Function you wish to debug. *This could be a Commerce function, such as Pricing, or a function on an action or attribute.*

```
2886        print myStringDict; print myIntegerDict; print myFloatDict; print masApprovalDict;
2887        masApprovalDict = util.mASApprovals2(myStringDict, myIntegerDict, myFloatDict, masApprovalDict);
```

c. Locate the function that contains the Util Function you wish to debug. *This could be a Commerce function, such as Pricing, or a function on an action or attribute.*

```
{enterpriseAccruals=No, discErrMsg=, totalEEs=150, partDesc=WFN Solutions, family=Processing, proc
essingRegionMAS_quote=0069 - Central Plains, serviceLevel=Technology Only, partNum=10-F0040Z}
{qtyApprovals=150}
{netperunitfee=217.0, maxProcDisc=0.0, maxSetupDisc=0.0, maxW2Disc=0.0, discountPercent=60.0,
 discountAmt=325.5}
{mASFinanceApproval=false, fSAAppForDisc=false, mASCompHRApproval=false, cobraAppForDisc=fals
e, approval=false, mASBenefitsApproval=false, mASDivisionVPApprover=false, mASApprovalReason=}
mASApprovalReason: MAS: Processing Discount over 50%: 10-F0040Z#
```

---

2. Create a test script for the Util Function.

a. The Util Function Debugger can use a test script for debugging within the function itself. The test script creates the variables and stores the information produced in **Step 1** into the variables that will be returned in the Util Function.

b. Create and initialize variables that are the same type as those passed through the Util Function. *For stings, integers, of floats, the initialization can be the actual value produced in **Step 1**.*

```
myStringDict = dict("string");
myIntegerDict = dict("integer");
myFloatDict = dict("float");
masApprovalDict = dict("string");
```

c. Populate the variables with the information obtained in Step 1. *This is mainly for dictionaries and arrays; they are populated using BMLs the standard way.*

```
put(myStringDict, "enterpriseAccruals", "no");
put(myStringDict, "discErrMsg", "");
put(myStringDict, "totalEEs", "150");
put(myStringDict, "partDesc", "WFN Solutions");
put(myStringDict, "family", "Processing");
put(myStringDict, "processingRegionMAS_quote", "0069 - Central Plains");
put(myStringDict, "serviceLevel", "Technology Only");
put(myStringDict, "partNum", "10-F0040Z");

put(myIntegerDict, "qtyApprovals", 150);

put(myFloatDict, "netperunitfee", 54.25);
put(myFloatDict, "maxProcDisc", 0.0);
put(myFloatDict, "maxSetupDisc", 0.0);
put(myFloatDict, "maxW2Disc", 0.0);
put(myFloatDict, "discountPercent", 90.0);
put(myFloatDict, "discountAmt", 488.25);

put(masApprovalDict, "maSFinanceApproval", "false");
put(masApprovalDict, "fSAAForDisc", "false");
put(masApprovalDict, "mASCompHRApproval", "false");
put(masApprovalDict, "cobraAppForDisc", "false");
put(masApprovalDict, "approval", "false");
put(masApprovalDict, "mASBenefitsApproval", "false");
put(masApprovalDict, "mASDivisionVPApprover", "false");
put(masApprovalDict, "mASApprovalReason", "");
```

d. Return the Util Function holding its parameters.

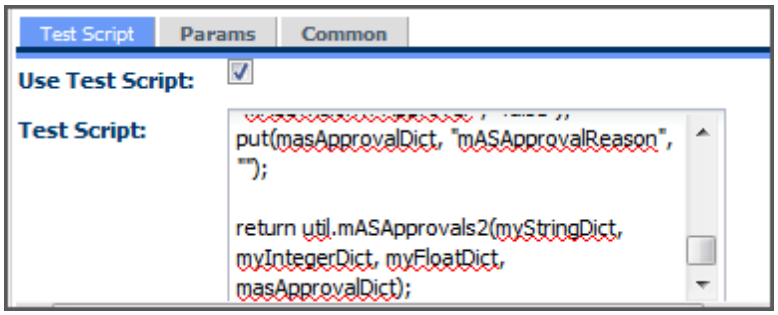
```
return util.mASApprovals2(myStringDict, myIntegerDict, myFloatDict, masApprovalDict);
```

---

3. Use the test script to generate the output.

   a. Copy and paste the test script produced in **Step 2** into the Test Script area of the Debugger.

   b. Select the **Use Test Script** checkbox.



c. Click Run to ensure that there are no errors in the test script. The return string will be generated here.



From here you can use the debugger like the Commerce function debugger. *You can change your values in your script as often as you like to test different scenarios and values.*

TROUBLESHOOTING

Think through how to correct an error with invalid data passed into a BML library, since it may be called from many different places. Follow proper error handling practices to ensure that the code is easy for others to maintain, and provides a good experience for end users.

NOTES



The same rules apply to both Util and Commerce libraries.

RELATED TOPICS

USING SOAP WITH BML

OVERVIEW



Before using SOAP as a solution, please check first that there isn't a way to do it with standard functionality.

ADMINISTRATION

1. Determine which type of SOAP call you need.
2. Navigate from: Admin > Web Services.
    - Record the Receiver URL (SOAP URL)
    - Generate a Sample SOAP Call and copy the sample data into a text editor
3. Replace all dynamic data with holder text.

    The holder text should be alpha-numeric in all CAPS, the only special characters allowed are underscores (_). In the example below SESSION_ID and TRANSACTION_ID are the holder text.

4. Example of `getTransaction` a Commerce SOAP API:

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soap env="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>FBA46BAE5D5DF082B4EC1B13E62DA32A</bm:sessionId></bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:schemaLocation>http://bigideas2010.bigmachines.com/bmfsweb/bigideas2010/schema/v1_0/commerce/quotes_process_bmClone_
24.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:transaction>
<bm:id/>
</bm:transaction>
```

```
</bm:getTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

5. Save file with .xml extension to File Manager under SOAP folder, make sure that the file name is descriptive. *In the example above, the file would be saved as getTransaction.xml.* It is important that the file is well formed.

6. In BML editor: *(This example gets a transaction and returns the SOAP call, this could be used in configuration)*.

```
/*Gathers quote fields and posts them externally, returns the XML response from the external system.*/
soapResponseXML = "";
//Prepare xml string to be used in WS request.
//Pull dynamic content from one or more quote and/or line level commerce attributes
dynamicQuoteContent="";
dynamicLineContent="";
//Populate the quote content directly
dynamicQuoteContent = "<quote:someQuoteAttribute1>" + someQuoteAttribute1_quote + "</quote:someQuoteAttribute1>"
+ "<quote:someQuoteAttribute2>" + someQuoteAttribute2_quote + "</quote:someQuoteAttribute2>" ;
//Loop over lines to populate the line content. Could use conditional logic to only use certain lines (not shown
here).
for line in line_process
{
dynamicLineContent=dynamicLineContent+"<item1:ItemNumber>"+someLineAttribute1_line+"</item1:ItemNumber>";
}
//Get a skeleton XML file that you've built used for this particular integration. Store it in the file manager
and access it from BML as follows.
getTransactionURL = "https://"+ lower(_system_supplier_company_name) +".bigmachines.com/bmfsweb/"+ lower(_system_
supplier_company_name) +"/image/folderName/xmlFileName.xml"; //XML SOAP Request file location
getTransactionFile = urldatabypost(getTransactionURL,"","",""); // Gets XML file.
//Replace placeholder content in the XML with the dynamic content gathered earlier.
getTransactionFile = replace(getTransactionFile,"$DYNAMICQUOTECONTENT$", dynamicQuoteContent);
getTransactionFile = replace(getTransactionFile,"$DYNAMICLINECONTENT$", dynamicLineContent);
//Set up the call to the external system
//Store the URL and the Action for the call in a data table to avoid URL hardcoding. Also makes the different
sites (dev, test, prod) interact with corresponding receiver URL's, and the URL's get passed around in site migra-
tions & refreshes without manual updating.
externalURL = "";
SOAPAction = "";
//Set up the site and the integration that's being performed
siteNameStr = lower(_system_supplier_company_name);
integrationType = "sampleType";
//Grab the URL and action name from the table. There should only be one corresponding row; this logic just takes
the first record.
```

```
soapURLRecs = bmql("select URL, SoapAction from SiteInfo where SiteName = $siteNameStr and Integration = $integrationType");

for eachRec in soapURLRecs{

externalURL = get(eachRec, "URL");

SOAPAction = get(eachRec, "SoapAction");

break;

}

//Put the SOAP Action into a dictionary for the post call

headerDict = dict("string");

put(headerDict, "SOAPAction", SOAPAction);

//Send the soap call and return the response to variable.

soapResponseXML= urldatabypost(externalURL, getTransactionFile ,"FAIL",headerdict);

return soapResponseXML;//The code that called this function can now parse this XML string and use its content for whatever it needs.
```

NOTES



CDATA tags are used when data should be ignored and not parsed. This is useful when the data contains special characters like: & and <.

For this example, the cdata isn't really needed because session id and transaction id should never contain special characters. Here they are used as a precaution, but when using other SOAP APIs they are necessary. `<![CDATA [data goes here] ]>`

RELATED TOPICS

BML CODING BEST PRACTICES

OVERVIEW

The purpose of good coding style is to write code that is readable. Readable code is easier to debug, extend, and modify, thus increasing your productivity and that of anyone else who will read your code. It is your duty as a programmer to adhere to style best practices.

Types of coding styles:

Structural Coding Style

1. The Function Header should describe the entire script's salient properties.

- The name of the script or function (for global BML search)
- Script location within BMI
- Parameters
- Return type
- Include short script descriptions or cautions for future programmers who might edit your code
- The name of the script author, date, function editors' names, and summarizing comments
  Example:

```
Name Date Comments
-------------------------------------------------------------------------------------------
John Doe 1/1/19 Created Function
Jane Doe 5/1/19 Added additional managers to notifications
```

2. Major sections of your code should be clearly marked and contain a brief explanation of its purpose, who wrote the code, and the date the code was written or modified.

- Provide visual contrast if possible
- Organize like chapters in a book or specific tasks in an algorithm
- Aids other programmers in identifying where they are in your script, and the current section's purpose
- Ensures your script is well structured and reminds you of what you did in different areas of a very large BML script (e.g. Commerce Pricing)
- Aids in quickly skimming or scanning BML script without having to read through multiple, individual lines of comments

3. Secondary sections of your code should also be clearly marked and contain a brief explanation, if necessary.

- Some major sections of your BML script can become very complex; decompose your steps and provide explanations for readers

4. Group related lines of code together.

- The proximity of related lines of code communicates to a reader that those lines are related
- This helps a reader identify which lines of code correspond to which pieces of functionality, therefore aiding: debugging, amendment, and extension

5. Follow conventional indentation and bracketing practices (alignment).
    - Use the Tab key in Notepad++
    - Every immediately nested code block should be indented exactly one tab to the right with respect to its parent content
    - Curly brackets should always open at the end of a line of code
        - *No code of any kind should follow on the same line after the opening of a curly bracket (which denotes the start of a nested code block)*
    - Curly brackets should always close at the beginning and end of a line of code
    - No code should exist on the same line as a closing curly bracket (which denotes the end of a nested code block)
        - *The only exception for the curly bracket rules should be the inner array declaration of a hard-coded 2D array*

Inline Coding Style

1. The name and format of a BML local variable should immediately communicate its purpose, intended behavior, and usage.

    Local variables used as constants should:
    - CAPITALIZE all letters
    - Use underscores to separate words in the name of the constant
        - E.g. `ITEM_DELIM = "*@*@*";`

    Local variables used as variables should:
    - The first letter should be lower case
    - The first letter of each subsequent word in the name should be UPPER CASE
    - DO NOT use underscores to delimit separate words in the name
        - E.g. `totalListPrice = 0.0`

    In all cases, the name of the variable should describe itself:
    - `totalListPrice` is the sum of all line item list prices

- Variable names for Arrays should have the "List", "Arr", "Array" and/or "2D" suffix
    - E.g. `docNumList`, `upgradedPartsArray`, `partsData2DArray`

Variable names for Dictionaries should have the "Dict" suffix, and have a comment at the end of the declaration explaining the mapping between Key and Value:

- E.g. `sequenceNumDict = dict(string) //Key: Document Number, Value: Sequence Number`
- E.g. `relatedLicenseToDocNumDict = dict(float) //Key: relatedLicense, Value: Document Number`
- The exception to this rule is when iterating over Records in a RecordSet
    - E.g. `"for partRecord in partRecords {"`

Variable names for RecordSets should have the "Records" suffix:

- E.g. `partRecords`

Variable names for Booleans should have the "is" or "has" prefix (answered with a "yes/no" answer):

- Possible exceptions include standard Booleans such as "debug"
    - E.g. `isUpgrade`, `isMandatory`, `hasRelatedLicense`

2. There should never be more than one statement on one line *(i.e. never more than one semicolon ending a statement)*.

3. Print statements should be extricated from convoluted code and separated into a debug-flag controlled if block, and if it doesn't cause the code to become less readable.

4. Functions with a single argument should always enclose that argument in parentheses e.g. `not(hasLicensing)` rather than `not hasLicensing`.

5. Extremely long single-line statements, such as String concatenations, should be split across multiple lines.



The repetition of these naming conventions will aid a reader in determining the purpose of any variable without having to refer to additional context.

RELATED TOPICS

BIGMACHINES QUERY LANGUAGE (BMQL)

OVERVIEW

BigMachines can query system tables and user-created data tables from within BML by using a SQL-like syntax. This functionality has replaced the deprecated `gettabledata()` and `getpartsdata()` functions. BMQL is a function that returns a new data type, `RecordSet`, containing the results of an SQL query. Click here for more information on SQL.

Highlights:

- A BMQL query does not cap the size of the results, unless it uses either Distinct or Order By; in this case, the result set can contain a maximum of 1000 records.
- Can access the parts database or user-defined tables.
- BMQL is context-sensitive and will recognize the language, currency, and selected price book depending on where the function is being invoked.

ADMINISTRATION

Basics

Syntax: RecordSet This data type is a collection of dictionaries. It can also be used as a function in conjunction with BMQL.

```
RecordSet recordset()
Behaviour/Example:
Returns a new RecordSet to be used for later assignments.
Example:
rs = recordset();
if(...) {
rs = bmql(query1);
}
else {
rs = bmql(query2);
}
return rs;
```

bmql(String sqlQuery) As mentioned above, the new BMQL function is based on SQL syntax. In SQL, most actions performed on a database are done with a statement like: `SELECT + FROM`. Similar to SQL, in BMQL, your "statement" would read: `SELECT (fields you want to return)FROM(which database you will be searching through)`. When the function syntax is written as: `bmql(String sqlQuery)`, the `sqlQuery` is really the "statement", similar to the example above. *The `sqlQuery` MUST be written as a String.*

```
3 results = bmql("SELECT part_number FROM _parts");
```

Statement Keywords:

Statement	Description	Example
SELECT	This is where you select your columns (or fields) of desired data.	select column 1, column 2, column 3, column 4
FROM	This is where you select the database object from which to retrieve the data.	from someUser-CreatedTable
DISTINCT	This keyword can be used with select to return distinct values, removing duplicates.	select distinct column 1, column 2, column 3, column 4
ORDER BY	This is used to sort the data returned on the provided fields. Be default, fields will be returned in ascending order. Use 'desc' to reverse the sort order to descending and use 'asc' to reverse the sort order to ascending.	order by column 1 desc

---

Parameters

When using BMQL, your `WHERE` clauses will be your parameters. BML variables can also be used in `WHERE` clause conditions.

```
3 var_1 = "part123";
4 lead = integer[]{1,2,3};
5
6 results = bmql("SELECT part_number FROM _parts WHERE part_number=$var_1 AND lead_time IN $lead")
```



$ symbol must precede the BML variable name when used in the query string *(as shown in line 6 above)*.

For the `IN` condition, the variables should be of data type: `string[]`, `integer[]` or `float[]`.



The variable data types that can be used are: String, Integer, and Float.



Line item variables cannot be used directly in the query. You can assign line item variables to variables in BML and then use them in the query string.

WHERE Condition

The condition is field operator . **For Example:** `part_number = 'BL-5C'`. *String values should be enclosed in single quotations marks, while float and integer values should be used without them (i.e., `valve_pressure= 1.235`).*

- *Logical operators can be used to group the conditions:*

```
3 value<500 AND (currency = 'USD' OR currency = 'EUR')
```

- *Parentheses can be used to change the precedence:*

```
3 value<500 AND currency = 'USD' OR currency = 'EUR'
```



The `WHERE` clause is not supported for BMQL Transaction.



This is a conditional evaluation where a Boolean value can be used with the `AND` operator to evaluate the predicate conditionally in the `WHERE` clause. Where `eval` is a Boolean variable in the script. When `eval` is `TRUE`, the predicate `field = value` is used in the criteria and when `eval` is `FALSE`, the predicate `field = value` is ignored. *The condition MUST come before the predicate.*

Transaction Data

This allows users to access Commerce transaction data, from both main document and sub-document, in advanced functions within Configuration. *This functionality is not supported outside of configuration.*

Data Available using BMQL Transaction:

- Date
- Integer
- Float
- String
- Currency
- Summation
- Boolean

The BMQL Transaction function is context sensitive. This implies the following:

- The function will automatically recognize the transaction ID when the user arrives on the Configuration page through Commerce. *It is not possible to query a separate transaction using this function.*
- Translations will be returned for menu values based on the current user's language. For other string type attribute values, the user-entered value will be returned as translations are not defined for such inputs.

- For currency attributes, the exchange rate will be applied on the returned value if the current user's session currency is not the same as the site's base currency.



When the user's session currency is different from the site's base currency, exchange rates are applied to currency values being returned from advanced functions in Configuration.

The fullaccessuser should take this into account when using BMQL Transaction in advanced functions, since BMQL also applies the exchange rate on the returned value.



Approval Comments, History, RTE, HTML, and File Attachment attributes are not available through BMQL Transaction.

Operators

- **Relational Operators:** =, <>, <, >, <=, >=, LIKE, NOT LIKE, IN, NOT IN, IS NULL, IS NOT NULL
- **Logical Operators:** AND and OR

Operator	Description	Example
LIKE	This is used as "contains".	SELECT part_number FROM _parts WHERE part_number **LIKE** $var_1
NOT LIKE	This is used as "does not contain".	SELECT part_number FROM _parts WHERE part_number **IS NOT LIKE** $var_1
IN	This is used to find a value within an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **IN** $lead_time
NOT IN	This is used to find values outside of an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **NOT IN** $lead_time
IS NULL	This is used to find null values in an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **IS NULL**
IS NOT NULL	This is used to find values that are not null in an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **IS NOT NULL**

Return Type: RecordSet



% is the wild card.

Three functions are available to retrieve data from the result set depending on the data type of the value being returned: get(string), getInt(integer), getFloat(float).



Within each Function Wizard, scroll down to the end of the **Behavior** section to find a link that opens a pop-up window that will show ALL accessible databases and data tables. Click on a database or data table name to see a list of column names and whether it can be used in the WHERE clause.



The BMQL query is parsed at compile time.

OrderBy and Distinct cannot be grouped together using AND or OR. The order of operations can also be changed using parentheses.

For security reasons, the query cannot be built dynamically and passed into BMQL. Dynamic values can be passed in the WHERE clause of the query by preceding the variable name with a "$".



There is currently only ONE system database that can be accessed: _parts. *Below you will find a list of the column names that can be accessed from the parts database:*

_parts	
Column Name	**Can use in WHERE clause**
custom_field1	Yes
custom_field2	Yes
custom_field3	Yes
custom_field4	Yes
custom_field5	Yes
custom_field6	Yes
custom_field7	Yes
custom_field8	Yes
custom_field9	Yes
custom_field10	Yes
custom_field11	Yes
custom_field12	Yes
custom_field13	Yes
custom_field14	Yes
custom_field15	Yes
custom_field16	No
custom_field17	No
custom_field18	Yes
custom_field19	Yes
custom_field20	Yes
custom_field21	Yes
custom_field22	Yes
custom_field23	Yes
custom_field24	Yes
custom_field25	Yes
custom_field26	Yes
custom_field27	Yes
custom_field28	Yes
custom_field29	Yes
custom_field30	Yes
part_number	Yes
description	No
lead_time	Yes
price	No
direct_buy	No
units	No

EXAMPLE USE CASES

For all of the use cases below, we'll be using a user created data table named "sammie". Click on a Sample Use Case below to see an example.

Data List

Select	Part	Type	Description	Price	Date Modified
☐	HS001	Hot	Meatball Sandwich	7.95	05/27/2010 3:38 PM
☐	HS002	Hot	Grilled Chicken Sandwich	8.95	05/27/2010 3:38 PM
☐	HS003	Hot	Italian Beef	6.95	05/27/2010 3:38 PM
☐	HS004	Hot	Grilled Veggie Sandwich	4.95	05/27/2010 3:38 PM
☐	CS001	Cold	Tuna Sandwich	5.5	05/27/2010 3:38 PM
☐	CS002	Cold	Bacon Turkey Club	7.65	05/27/2010 3:38 PM
☐	CS003	Cold	Egg Salad Sandwich	8.15	05/27/2010 3:38 PM
☐	CS004	Cold	Cold Cuts Collide	6.49	05/27/2010 3:38 PM

Sample Use Case 1: Using BMQL function

In this example, let's say you'd like to run a query to return the **Price** and **Type** from the "sammie" table shown above.

1. Using the information from the "sammie" data table, create a new script using BMQL that will return Price and Type. *Click the checkboxes in the **Select** column to select your columns and **From** to select your database object.*

2. Create a `for...loop` to loop through your record set created using the BMQL function.

3. In this example, we are using the print statement to show the results of the query.

```
1 results = bmql("SELECT Price, Type FROM sammie");
2 for record in results{
3         print record;
4
5 return "";
```

Now, you can compare the results of the query to what is on the table and see that it pulled the correct **Type** and **Price** from the table.

Console

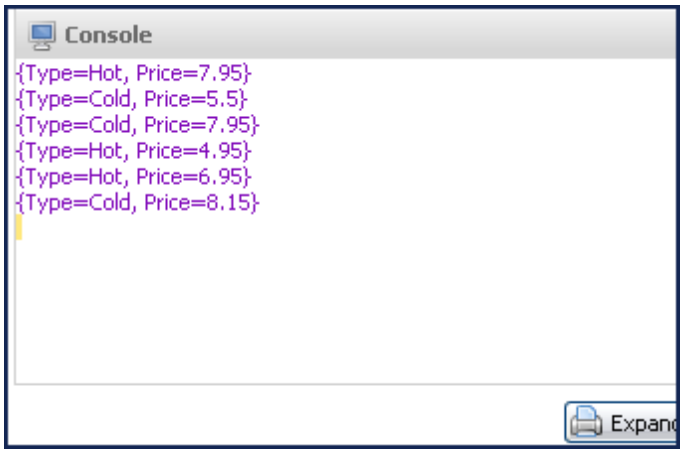
```
{Type=Hot, Price=7.95}
{Type=Hot, Price=7.95}
{Type=Hot, Price=6.95}
{Type=Hot, Price=4.95}
{Type=Cold, Price=5.5}
{Type=Cold, Price=7.95}
{Type=Cold, Price=8.15}
{Type=Cold, Price=7.95}
```

Expand

In this example, we are using the same sample case, but adding the "distinct" keyword. Remember, using distinct will only return distinct values, essentially removing any duplicates.

```
recordSet = bmql("select distinct Price, Type from sammie");
for record in recordSet{
        print record;
}
return "";
```

As you can see, where there were multiple sets that were the same in the first example, now those have been removed.



Sample Use Case 2: Using the get function and the WHERE Clause

This example takes Sample Use Case 1 a step further by adding a `WHERE` clause to the select statement. We also add the `get()` function.

1. Using the information from the "sammie" data table, create a new script using BMQL that will return **Part**, **Description** and **Price** when the **Type** in the data table is "Hot". *Click the checkboxes in the **Select** column to select your columns and **From** to select your database object.*

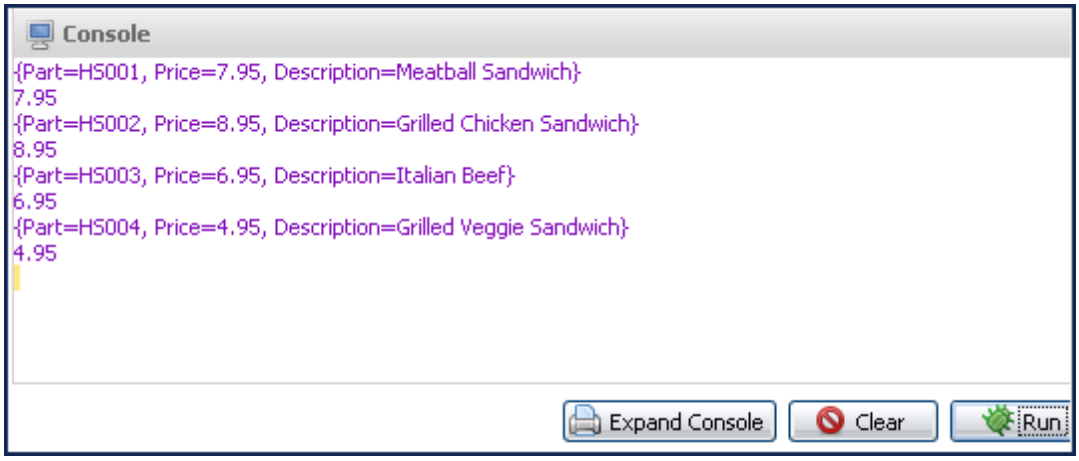
2. Create a `for...loop` to go through your record set created using the BMQL function.

3. Use the `get()` function with record being used like a dictionary and **Price** being the data to be returned.

```
results = bmql("SELECT Part, Description, Price FROM sammie WHERE Type ='Hot'");
for record in results{
        print record;
        temp = get(record, "Price");
                print temp;
}

return "";
```

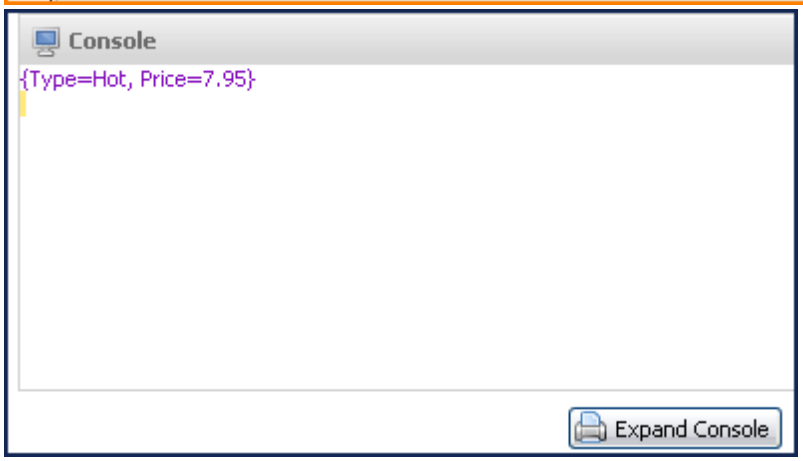
As you can see below, "record" acts like a dictionary and returns the columns and data you requested. You should also notice the `get()` function at work, returning the price for each of the records.



Sample Use Case 3: Using a WHERE condition

In this example, we are going to add a condition. So, the first screen show we'll say that if the condition is true, that the predicate will query for the part number "HS001". *When adding a* `WHERE` *condition, the syntax is* `$eval AND field = value`. In this case, we are searching for part number "HS001" when the condition is True.

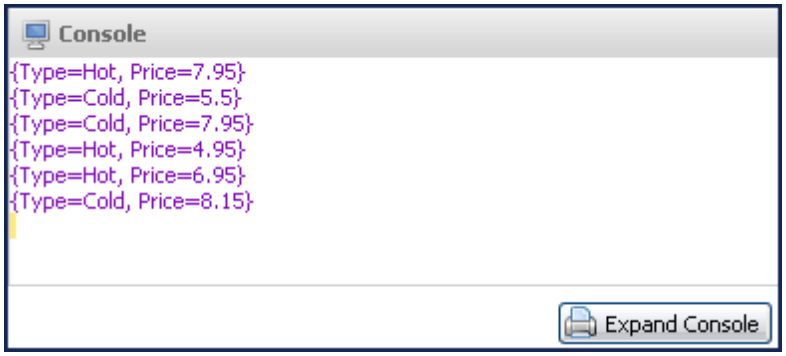
```
1 eval1 = true;
2 eval2 = false;
3
4 recordSet = bmql("select distinct Price, Type
5                   from sammie where $eval1 AND Part = 'HS001'");
6 for record in recordSet{
7         print record;
8 }
9 return "";
```



In the console, you'll notice that when the part number is "HS001", the **Price** and **Type** have been returned. We'll then evaluate what happens if the condition is False:

```
eval1 = true;
eval2 = false;

recordSet = bmql("select distinct Price, Type
                  from sammie where $eval2 AND Part = 'HS001'");
for record in recordSet{
        print record;
}
return "";
```



In the console, you'll notice that when the condition is False, the predicate (`Part = 'HS001'`) is ignored and all other results are returned.

Sample Use Case 4: Errors

The user can retrieve errors or warnings using the `getMessage` function.

```
value = "1";
results = bmql("SELECT part_number, price FROM _parts WHERE part_number = $value");
for record in results{
        if (hasError(results)){
                print (getMessage(results));
        }
}

return "";
```

Sample Use Case 5: Recommended Item Rule

```
//Recommended Items > Hot Sandwiches
//Data Table:  sammie

//Set up the return string
final = "";
delimiterItem = "~";
delimiterEnd = "|^|";

//Create the query using bmql
results = bmql("select distinct Part, Price, Description from sammie");

//Loop through the record set and populate the return string
for record in results{
        print record;
        final = final + get(record,"Part") + delimiterItem + "1" +
        delimiterItem + get(record, "Description") + delimiterItem +
        get(record,"Price") + delimiterEnd;
}

return final;
```

Sample Use Case 6: Using BMQL Transaction

In this example, we'll show you how to return commerce transaction data back to configuration. *The attribute Opportunity Name has been set with the value Toni's Pizza.*



This was set through a recommendation rule using BMQL Transaction. Notice the use of `commerce.quote_process` after the `FROM` statement. This is the variable name of the quote document from where you're querying data. You can also use the variable name of the sub-document (i.e. `line_process`).

```
opty = "";
records = BMQL("select opportunityName_quote from commerce.quote_process");
        for result in records {
                opty = get(result, "opportunityName_quote");
        }
return opty;
```



The WHERE clause is not accepted when you are returning commerce data to configuration.

NOTES



BMQL is case-sensitive. *Be aware of this when selecting your columns, fields, and database objects.*

CAPITALIZE your keywords to ensure that your query string easy to read.

For debugging BMQL Transaction, use the variable bsId(transaction ID).

Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.*



On reconfigure, BMQL Transaction will return all line items.



The WHERE clause is not supported for BMQL transaction.

If the user does not arrive through Commerce, the BMQL transaction will return null. *This is also occurs when a model is configured using SOAP.*

RELATED TOPICS

BIGMACHINES QUERY LANGUAGE (BMQL)

OVERVIEW

BigMachines can query system tables and user-created data tables from within BML by using a SQL-like syntax. This functionality has replaced the deprecated `gettabledata()` and `getpartsdata()` functions. BMQL is a function that returns a new data type, `RecordSet`, containing the results of an SQL query. Click here for more information on SQL.

Highlights:

- A BMQL query does not cap the size of the results, unless it uses either Distinct or Order By; in this case, the result set can contain a maximum of 1000 records.
- Can access the parts database or user-defined tables.
- BMQL is context-sensitive and will recognize the language, currency, and selected price book depending on where the function is being invoked.

ADMINISTRATION

Basics

Syntax: RecordSet This data type is a collection of dictionaries. It can also be used as a function in conjunction with BMQL.

```
RecordSet recordset()
Behaviour/Example:
Returns a new RecordSet to be used for later assignments.
Example:
rs = recordset();
if(...) {
rs = bmql(query1);
}
else {
rs = bmql(query2);
}
return rs;
```

bmql(String sqlQuery) As mentioned above, the new BMQL function is based on SQL syntax. In SQL, most actions performed on a database are done with a statement like: `SELECT + FROM`. Similar to SQL, in BMQL, your "statement" would read: `SELECT (fields you want to return)FROM(which database you will be searching through)`. When the function syntax is written as: `bmql(String sqlQuery)`, the `sqlQuery` is really the "statement", similar to the example above. *The `sqlQuery` MUST be written as a String.*

```
3 results = bmql("SELECT part_number FROM _parts");
```

Statement Keywords:

Statement	Description	Example
SELECT	This is where you select your columns (or fields) of desired data.	select column 1, column 2, column 3, column 4
FROM	This is where you select the database object from which to retrieve the data.	from someUser-CreatedTable
DISTINCT	This keyword can be used with select to return distinct values, removing duplicates.	select distinct column 1, column 2, column 3, column 4
ORDER BY	This is used to sort the data returned on the provided fields. Be default, fields will be returned in ascending order. Use 'desc' to reverse the sort order to descending and use 'asc' to reverse the sort order to ascending.	order by column 1 desc

---

Parameters

When using BMQL, your `WHERE` clauses will be your parameters. BML variables can also be used in `WHERE` clause conditions.

```
3 var_1 = "part123";
4 lead = integer[]{1,2,3};
5
6 results = bmql("SELECT part_number FROM _parts WHERE part_number=$var_1 AND lead_time IN $lead")
```



$ symbol must precede the BML variable name when used in the query string *(as shown in line 6 above)*.

For the `IN` condition, the variables should be of data type: `string[]`, `integer[]` or `float[]`.



The variable data types that can be used are: String, Integer, and Float.



Line item variables cannot be used directly in the query. You can assign line item variables to variables in BML and then use them in the query string.

WHERE Condition

The condition is field operator . **For Example:** `part_number = 'BL-5C'`. *String values should be enclosed in single quotations marks, while float and integer values should be used without them (i.e., `valve_pressure= 1.235`).*

- *Logical operators can be used to group the conditions:*

```
3 value<500 AND (currency = 'USD' OR currency = 'EUR')
```

- *Parentheses can be used to change the precedence:*

```
3 value<500 AND currency = 'USD' OR currency = 'EUR'
```



The `WHERE` clause is not supported for BMQL Transaction.



This is a conditional evaluation where a Boolean value can be used with the `AND` operator to evaluate the predicate conditionally in the `WHERE` clause. Where `eval` is a Boolean variable in the script. When `eval` is `TRUE`, the predicate `field = value` is used in the criteria and when `eval` is `FALSE`, the predicate `field = value` is ignored. *The condition MUST come before the predicate.*

Transaction Data

This allows users to access Commerce transaction data, from both main document and sub-document, in advanced functions within Configuration. *This functionality is not supported outside of configuration.*

Data Available using BMQL Transaction:

- Date
- Integer
- Float
- String
- Currency
- Summation
- Boolean

The BMQL Transaction function is context sensitive. This implies the following:

- The function will automatically recognize the transaction ID when the user arrives on the Configuration page through Commerce. *It is not possible to query a separate transaction using this function.*
- Translations will be returned for menu values based on the current user's language. For other string type attribute values, the user-entered value will be returned as translations are not defined for such inputs.

- For currency attributes, the exchange rate will be applied on the returned value if the current user's session currency is not the same as the site's base currency.



When the user's session currency is different from the site's base currency, exchange rates are applied to currency values being returned from advanced functions in Configuration.

The fullaccessuser should take this into account when using BMQL Transaction in advanced functions, since BMQL also applies the exchange rate on the returned value.



Approval Comments, History, RTE, HTML, and File Attachment attributes are not available through BMQL Transaction.

Operators

- **Relational Operators:** =, <>, <, >, <=, >=, LIKE, NOT LIKE, IN, NOT IN, IS NULL, IS NOT NULL
- **Logical Operators:** AND and OR

Operator	Description	Example
LIKE	This is used as "contains".	SELECT part_number FROM _parts WHERE part_number **LIKE** $var_1
NOT LIKE	This is used as "does not contain".	SELECT part_number FROM _parts WHERE part_number **IS NOT LIKE** $var_1
IN	This is used to find a value within an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **IN** $lead_time
NOT IN	This is used to find values outside of an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **NOT IN** $lead_time
IS NULL	This is used to find null values in an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **IS NULL**
IS NOT NULL	This is used to find values that are not null in an array.	SELECT part_number FROM _parts WHERE part_number LIKE $var_1 **IS NOT NULL**

Return Type: RecordSet



% is the wild card.

Three functions are available to retrieve data from the result set depending on the data type of the value being returned: get(string), getInt(integer), getFloat(float).



Within each Function Wizard, scroll down to the end of the **Behavior** section to find a link that opens a pop-up window that will show ALL accessible databases and data tables. Click on a database or data table name to see a list of column names and whether it can be used in the WHERE clause.



The BMQL query is parsed at compile time.

OrderBy and Distinct cannot be grouped together using AND or OR. The order of operations can also be changed using parentheses.

For security reasons, the query cannot be built dynamically and passed into BMQL. Dynamic values can be passed in the WHERE clause of the query by preceding the variable name with a "$".



There is currently only ONE system database that can be accessed: _parts. *Below you will find a list of the column names that can be accessed from the parts database:*

_parts	
Column Name	**Can use in WHERE clause**
custom_field1	Yes
custom_field2	Yes
custom_field3	Yes
custom_field4	Yes
custom_field5	Yes
custom_field6	Yes
custom_field7	Yes
custom_field8	Yes
custom_field9	Yes
custom_field10	Yes
custom_field11	Yes
custom_field12	Yes
custom_field13	Yes
custom_field14	Yes
custom_field15	Yes
custom_field16	No
custom_field17	No
custom_field18	Yes
custom_field19	Yes
custom_field20	Yes
custom_field21	Yes
custom_field22	Yes
custom_field23	Yes
custom_field24	Yes
custom_field25	Yes
custom_field26	Yes
custom_field27	Yes
custom_field28	Yes
custom_field29	Yes
custom_field30	Yes
part_number	Yes
description	No
lead_time	Yes
price	No
direct_buy	No
units	No

EXAMPLE USE CASES

For all of the use cases below, we'll be using a user created data table named "sammie". Click on a Sample Use Case below to see an example.

Data List

Select	Part	Type	Description	Price	Date Modified
☐	HS001	Hot	Meatball Sandwich	7.95	05/27/2010 3:38 PM
☐	HS002	Hot	Grilled Chicken Sandwich	8.95	05/27/2010 3:38 PM
☐	HS003	Hot	Italian Beef	6.95	05/27/2010 3:38 PM
☐	HS004	Hot	Grilled Veggie Sandwich	4.95	05/27/2010 3:38 PM
☐	CS001	Cold	Tuna Sandwich	5.5	05/27/2010 3:38 PM
☐	CS002	Cold	Bacon Turkey Club	7.65	05/27/2010 3:38 PM
☐	CS003	Cold	Egg Salad Sandwich	8.15	05/27/2010 3:38 PM
☐	CS004	Cold	Cold Cuts Collide	6.49	05/27/2010 3:38 PM

Sample Use Case 1: Using BMQL function

In this example, let's say you'd like to run a query to return the **Price** and **Type** from the "sammie" table shown above.

1. Using the information from the "sammie" data table, create a new script using BMQL that will return Price and Type. *Click the checkboxes in the **Select** column to select your columns and **From** to select your database object.*

2. Create a `for...loop` to loop through your record set created using the BMQL function.

3. In this example, we are using the print statement to show the results of the query.

```
1 results = bmql("SELECT Price, Type FROM sammie");
2 for record in results{
3         print record;
4
5 return "";
```

Now, you can compare the results of the query to what is on the table and see that it pulled the correct **Type** and **Price** from the table.

Console

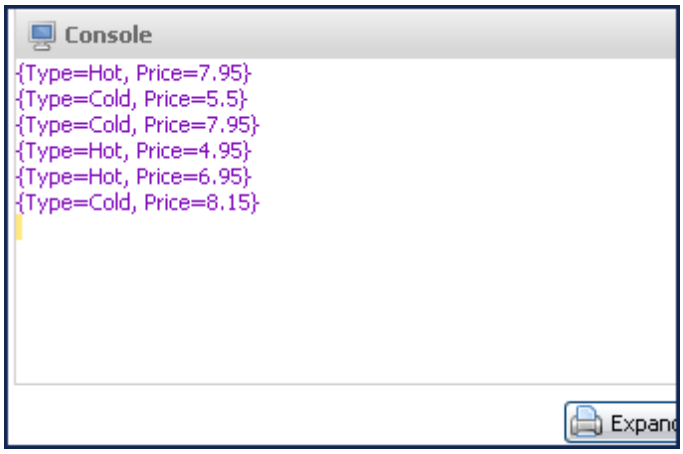
```
{Type=Hot, Price=7.95}
{Type=Hot, Price=7.95}
{Type=Hot, Price=6.95}
{Type=Hot, Price=4.95}
{Type=Cold, Price=5.5}
{Type=Cold, Price=7.95}
{Type=Cold, Price=8.15}
{Type=Cold, Price=7.95}
```

Expand

In this example, we are using the same sample case, but adding the "distinct" keyword. Remember, using distinct will only return distinct values, essentially removing any duplicates.

```
recordSet = bmql("select distinct Price, Type from sammie");
for record in recordSet{
        print record;
}
return "";
```

As you can see, where there were multiple sets that were the same in the first example, now those have been removed.



Sample Use Case 2: Using the get function and the WHERE Clause

This example takes Sample Use Case 1 a step further by adding a `WHERE` clause to the select statement. We also add the `get()` function.

1. Using the information from the "sammie" data table, create a new script using BMQL that will return **Part**, **Description** and **Price** when the **Type** in the data table is "Hot". *Click the checkboxes in the **Select** column to select your columns and **From** to select your database object.*

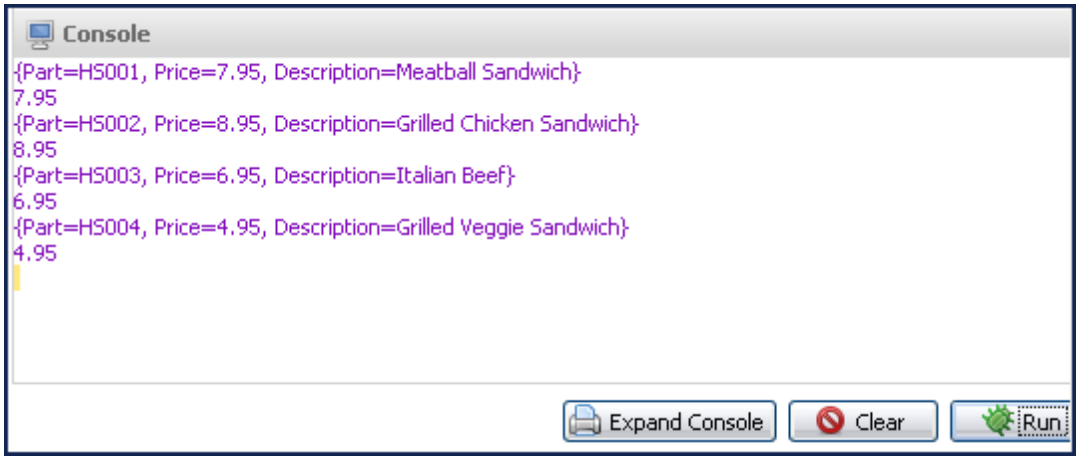
2. Create a `for...loop` to go through your record set created using the BMQL function.

3. Use the `get()` function with record being used like a dictionary and **Price** being the data to be returned.

```
results = bmql("SELECT Part, Description, Price FROM sammie WHERE Type ='Hot'");
for record in results{
        print record;
        temp = get(record, "Price");
                print temp;
}

return "";
```

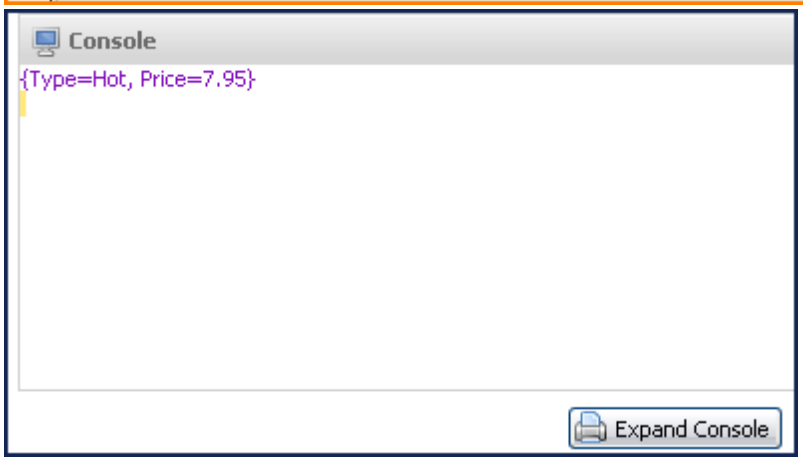
As you can see below, "record" acts like a dictionary and returns the columns and data you requested. You should also notice the `get()` function at work, returning the price for each of the records.



Sample Use Case 3: Using a WHERE condition

In this example, we are going to add a condition. So, the first screen show we'll say that if the condition is true, that the predicate will query for the part number "HS001". *When adding a* `WHERE` *condition, the syntax is* `$eval AND field = value`. In this case, we are searching for part number "HS001" when the condition is True.

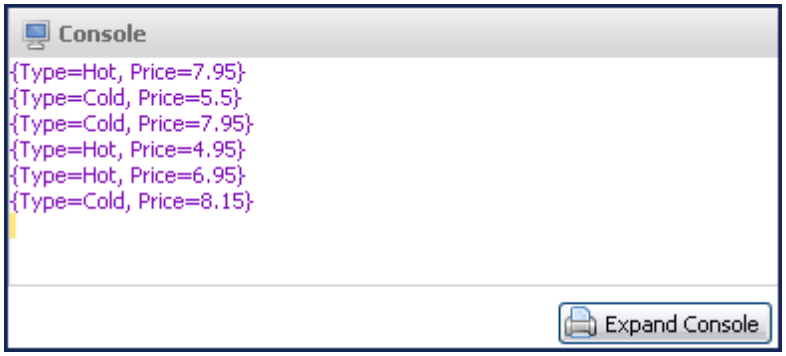
```
1 eval1 = true;
2 eval2 = false;
3
4 recordSet = bmql("select distinct Price, Type
5                   from sammie where $eval1 AND Part = 'HS001'");
6 for record in recordSet{
7         print record;
8 }
9 return "";
```



In the console, you'll notice that when the part number is "HS001", the **Price** and **Type** have been returned. We'll then evaluate what happens if the condition is False:

```
eval1 = true;
eval2 = false;

recordSet = bmql("select distinct Price, Type
                  from sammie where $eval2 AND Part = 'HS001'");
for record in recordSet{
        print record;
}
return "";
```



In the console, you'll notice that when the condition is False, the predicate (`Part = 'HS001'`) is ignored and all other results are returned.

Sample Use Case 4: Errors

The user can retrieve errors or warnings using the `getMessage` function.

```
value = "1";
results = bmql("SELECT part_number, price FROM _parts WHERE part_number = $value");
for record in results{
        if (hasError(results)){
                print (getMessage(results));
        }
}

return "";
```

Sample Use Case 5: Recommended Item Rule

```
//Recommended Items > Hot Sandwiches
//Data Table:  sammie

//Set up the return string
final = "";
delimiterItem = "~";
delimiterEnd = "|^|";

//Create the query using bmql
results = bmql("select distinct Part, Price, Description from sammie");

//Loop through the record set and populate the return string
for record in results{
        print record;
        final = final + get(record,"Part") + delimiterItem + "1" +
        delimiterItem + get(record, "Description") + delimiterItem +
        get(record,"Price") + delimiterEnd;
}

return final;
```

Sample Use Case 6: Using BMQL Transaction

In this example, we'll show you how to return commerce transaction data back to configuration. *The attribute Opportunity Name has been set with the value Toni's Pizza.*



This was set through a recommendation rule using BMQL Transaction. Notice the use of `commerce.quote_process` after the `FROM` statement. This is the variable name of the quote document from where you're querying data. You can also use the variable name of the sub-document (i.e. `line_process`).

```
opty = "";
records = BMQL("select opportunityName_quote from commerce.quote_process");
        for result in records {
                opty = get(result, "opportunityName_quote");
        }
return opty;
```



The WHERE clause is not accepted when you are returning commerce data to configuration.

NOTES



BMQL is case-sensitive. *Be aware of this when selecting your columns, fields, and database objects.*

CAPITALIZE your keywords to ensure that your query string easy to read.

For debugging BMQL Transaction, use the variable bsId(transaction ID).

Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.*



On reconfigure, BMQL Transaction will return all line items.



The WHERE clause is not supported for BMQL transaction.

If the user does not arrive through Commerce, the BMQL transaction will return null. *This is also occurs when a model is configured using SOAP.*

RELATED TOPICS

ARRAYS & ARRAY FUNCTIONS OVERVIEW

OVERVIEW

BigMachines declares, initializes, populates, and manipulates arrays using BML. An array is an ordered collection of values, referenced by a single variable name. Think of an array as a table, with values in rows and columns. Another way to think of arrays is in relation to an x-y axis. If you recall, data sets are formatted as (x, y). Data in arrays is similar:

```
(row, column)
(x, y)
("color", "shape")
```

1-D Array

	Name	Name	Name	Name	Name	Name	Name	Name	Name
Variable	1	2	3	4	5	6	7	8	9
Index	0	1	2	3	4	5	6	7	8

The index indicates the position of the value within the array. So, if we were looking for Value 7 within the array, its location is Index 6. Indexes are zero-based in BML, meaning that the first element will always have an index of 0 and the last element will have an index of i(number of element). For more information, see 1-D Arrays.

2-D Array

There are also 2-dimensional arrays. These arrays can be thought of as a grid or a table with multiple rows and columns. The convention for defining and accessing these arrays is [row, column]. For more information, see 2-D Arrays.

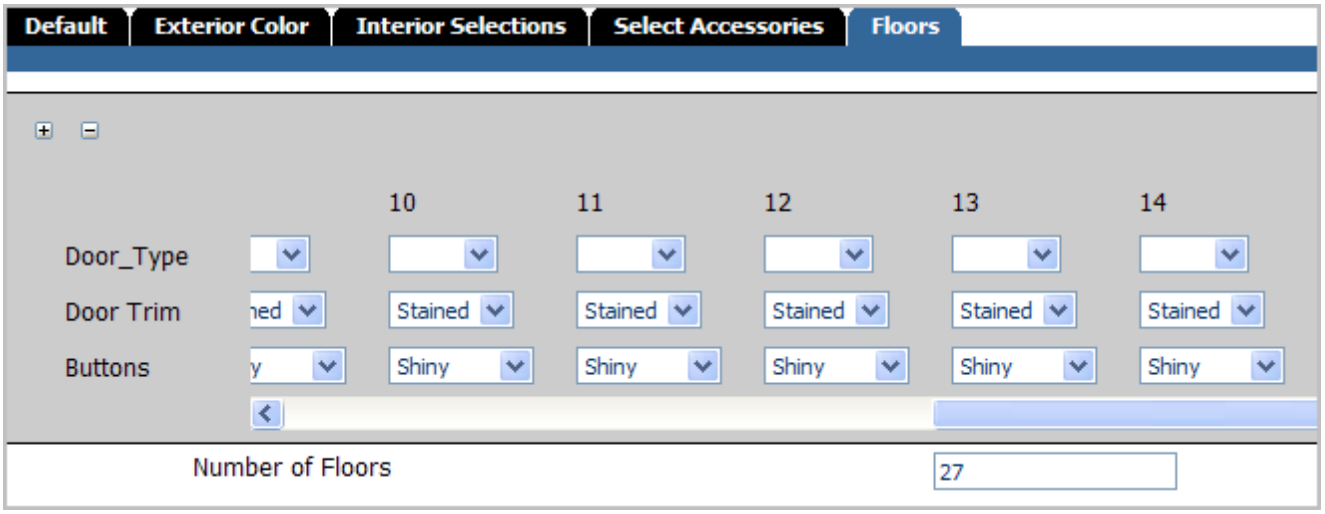
	Column 1	Column 2	Column 3
Variable 1	2	6	5
Variable 2	7	5	1
Variable 3	8	4	3

ADMINISTRATION

Array Attributes

What are array attributes? Array attributes are used to allow users to easily group multiple, scalar attributes using a single data structure and are available both for the selection and configuration process. Similar to other configurable attributes, you can create array attributes at all levels in the configuration hierarchy, but can delete them only at the level in which they were created. You can create array attributes for all the available data types (Float, Integer, String).

Example of an array attribute (user-side): In the example below, the user had selected that three floors needed to have their elevators configured. From this one screen, you can individually configure each elevator. As you can see, each one has a different combination of door type, door trim and buttons.



Control Attribute: This size of arrays are defined by the control attribute. When creating a control attribute, the attribute itself must have a Data Type of integer and you must make sure that the Array Type box is NOT checked. In the case below, the user wants to configure 27 floors. Number Of Floors is the control attribute.

Before You Begin:

Here is some general information regarding arrays that you should know before you get started. *Another way to think of an array is as a 1-D or 2-D table.*

1-D table of an array attribute

Elevator #	1	2	3
Door_Type	Stone	Blue	Red

2-D table of an array attribute

Elevator #	1	2	3
Door_Type	Stone	Blue	Red
Door_Trim	Painted	White	Stained
Buttons	Shiny	Easy	Plastic

BML Array Syntax

Element	Meaning
[]	Indicates a 1-D array
[] []	Indicates a 2-D array
[n]	1-D array with a defined size
[n][n]	2-D array with a defined size

Element	Meaning
{x,y}	Elements within a 1-D array
{(x,y), (z,r)}	Elements within a 2-D array

Indexes

Index values in a 1-D array

ROW 1	Index 0	Index 1	Index 2	Index 3	Index 4	Index 5	Index 6	Index 7

Index values in a 2-D array

Index values represent both the row index and the column index, so the basic syntax is `index[row][column]`.

	COL 1	COL 2	COL 3
ROW 1	Index [0][0]	Index [0][1]	Index [0][2]
ROW 2	Index [1][0]	Index [1][1]	Index [1][2]
ROW 3	Index [2][0]	Index [2][1]	Index [2][2]

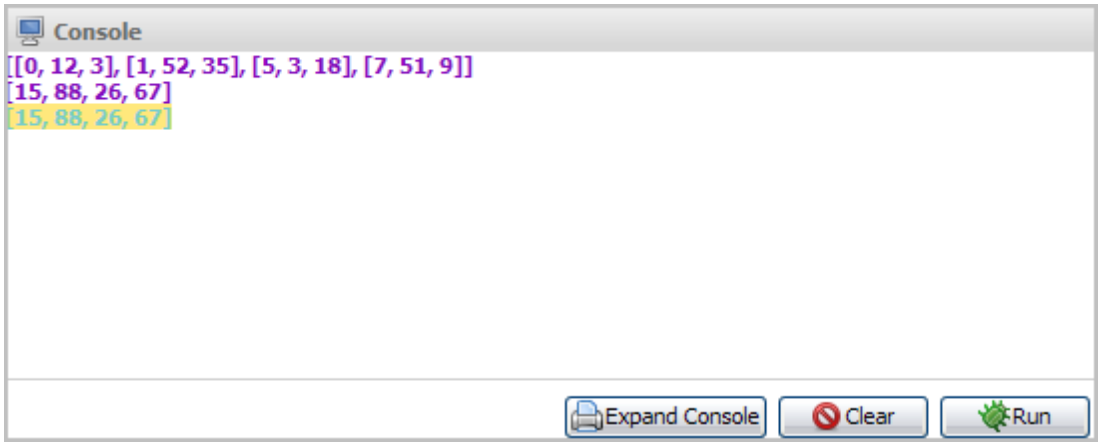
Populating Arrays

Examples of how to populate 1-D and 2-D arrays using a `for...loop`.

1-D Arrays

Create a 1-D array of values equal to the sum of each array element:

```
1  //Let's populate a 1-D array by looping through a 2-D array.
2  //Declare a 2-D integer array.
3  myArray = integer[][]{{0,12,3},{1,52,35},{5,3,18},{7,51,91}};
4          print myArray;
5  //Declare your empty 1-D array.
6  totals = integer[];
7  //Create a variable to represent the index.
8  index = 0;
9  //Begin looping through the 2-D array.
10 for row in myArray{ //Start looping through rows.
11         totals[index] = 0; //This is done to ensure your array begins populating at index=0
12         for col in row{ //Populate your empty array.
13                 totals[index] = totals[index] + col;
14         }
15 //You must increase your index to continue looping.
16 //Otherwise, values for the last element that is looped though will be found. In this case,
17         index = index +1;
18 }
19 print totals;
20 return totals;
```

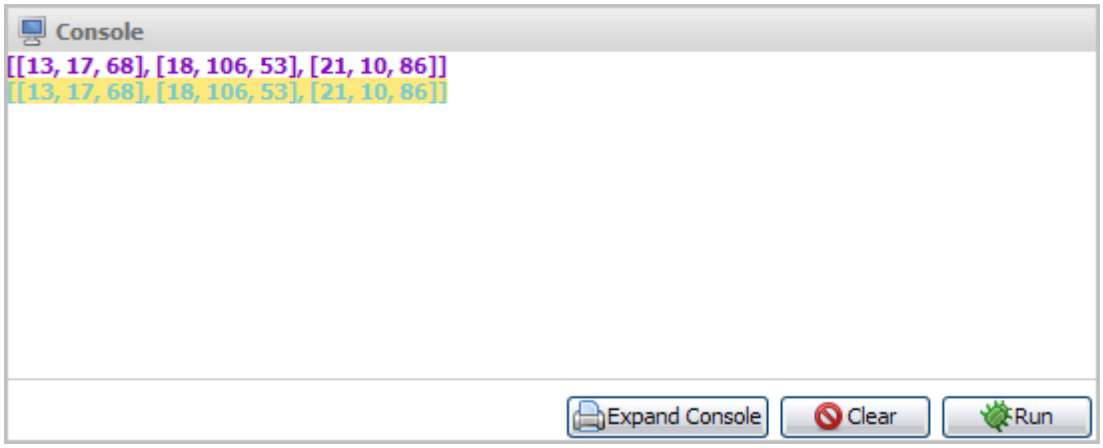


2-D Arrays

In this example, we are going to combine two 2-D arrays by adding together their value adn placing them in a new 2-D array. Remember that range() creates an integer array of indexes. We will be using two here.

```
1  //Populating a 2-D array
2  myArray = integer[][]{{1,2,3},{3,41,5},{6,1,8}};
3  secondArray = integer[][]{{12,15,65},{15,65,48},{15,9,78}};
4
5  colRange = range(3);
6  rowRange = range(3);
7  newArray = integer[][];
8
9  for row in rowRange{
10         for col in colRange{
11                 newArray[row][col] = myArray[row][col] + secondArray[row][col];
12         }
13 }
14
15 print newArray;
16 return newArray;
```

What we are going to do here is add together the values of the two arrays based on their indexes. `colRange` and `rowRange` are initializing integer arrays. So, both `colRange` and `rowRange = [0, 1, 2]`. respectively. Next, we have to declare and initialize the 2-D array that we are going to populate. Then, comes the `for...loop`. The first `for...loop` will loop through the rows of indexes in `rowRange`. The second will loop through the columns of indexes and thus will populate the new array by pulling out the column values.



RELATED TOPICS

ARRAY FUNCTIONS

OVERVIEW

An array is an ordered collection of values references by a single variable name. You could think of an array as a table, with values in rows and columns. Another way to think of arrays is in relation to an x-y axis. If you recall, data sets are formatted as (x, y). **Data in arrays is similar:**

```
(row, column)

(x,y)

("color", "shape")
```

ADMINISTRATION

append(arrayID, newArrayElem)

The function `append()` will attach a new element to the end of an array and can be used with initialized and uninitialized arrays. *Append works with 1-D arrays ONLY.*

#	Parameter	Data Type	Description
1	arrayIdentifier	string[], integer[], float[]	This is the given array that you are going to add an element to.
2	newArrayElem	float, float[], integer, integer[], string, string[], boolean, boolean[]	This represents the new element that you are appending to the array.

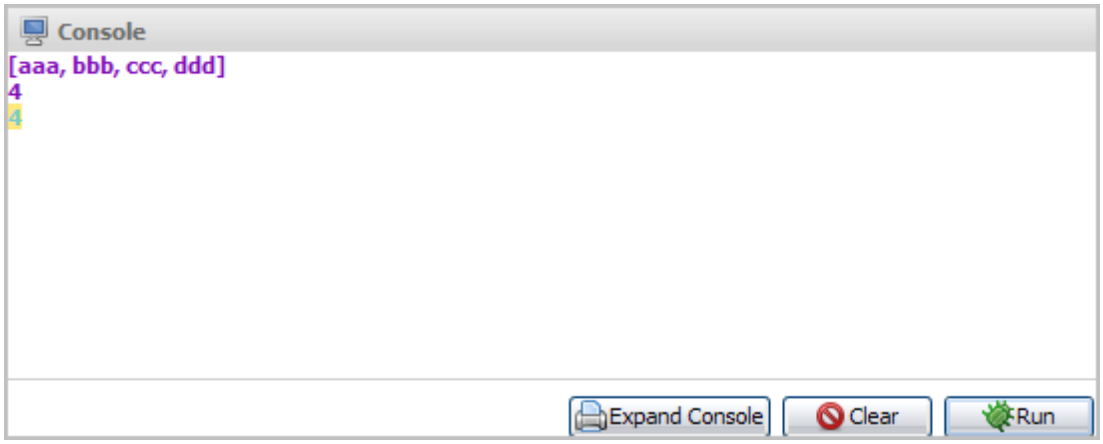
Return Type: Integer *(denotes the new size of the array)*



If array max size is reached and you try to append a new element, it will fail.

Example of append(arrayIdentifier, newArrayElem)

```
1 myArray=string[]{"aaa", "bbb", "ccc"};
2
3 results=append(myArray, "ddd");
4
5 print myArray;
6
7 return results;
```



If a null element is added to an array, it should be allowed to be added and the value of the element in the array should be null.

---

findinarray(arrayID, element)

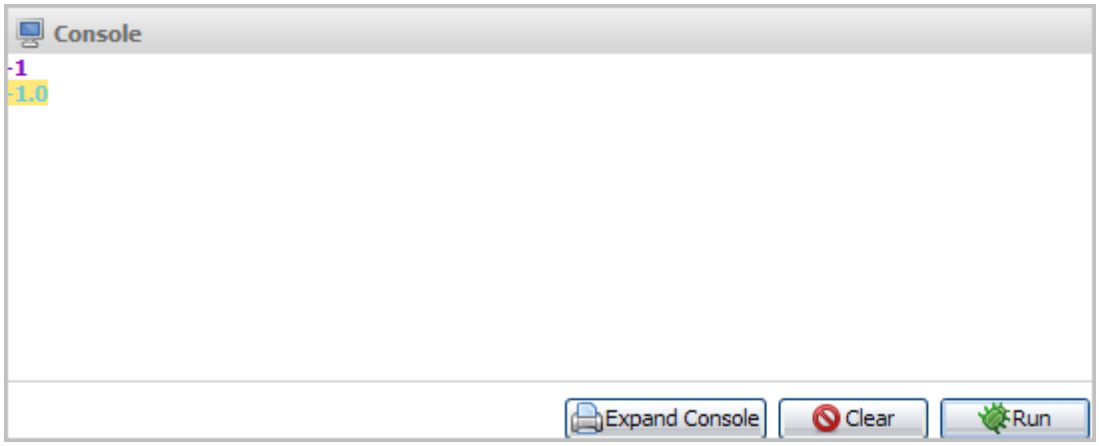
This function is used to check whether a certain element exists in an array. If it does, the index is returned, otherwise a -1 is returned. `Findinarray()` can only be used with 1-D arrays.

#	Parameter	Data Type	Description
1	arrayID	string[], integer[], float[]	This is the given array you are searching.
2	element	string, integer, float	Represents the element you are looking for in the given array.

Return Type: Integer *(denotes the index of the element in the array)*

Example of findinarray(arrayID, element)

```
1 myArray = float[]{123.15,123.54,123.085,123.09};
2
3 result = findinarray(myArray, 123.65);
4
5 print result;
6 return result;
```



---

isempty(arrayIdentifier)

This function determines if the array is empty. `Isempty()` can only be used with 1-D arrays.

#	Parameter	Data Type	Description
1	arrayIdentifier	string[], integer[], float[]	This is the given array.

Return Type: Boolean *(True if array is empty and False if it is not)*

Example of isempty(arrayIdentifier)

```
1 myArray = string[];
2 result = isempty(myArray);
3
4 print result;
5 return result;
```

The string array above is empty, therefore TRUE is returned.

---

join(str_Array, delimiter)

Concatenates a string array into a string with a specified identifier.



This function is case-sensitive.

#	Parameter	Data Type	Description
1	str_Array	string[]	This is the given array.
2	delimiter	string	Represents a delimiter to use when concatenating string elements.

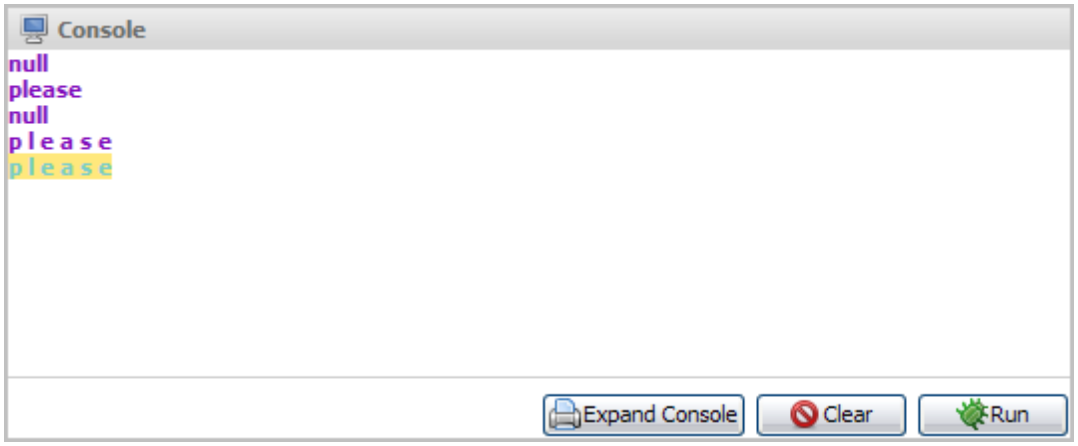
Return Type: String

If str array is empty, then the function will return an empty string. If BOTH str and delimiter are empty, then the function will return an empty string.

Example of join(str_Array, delimiter)

The following example will be similar to `split()`. You will see what happens when either the `str_Array` or delimiter are empty or if one is and one is not. The last example shows what happens when both the `str_Array` and delimiter have values.

```
1  myArray=string[]{"p","l","e","a","s","e"};
2  emptyArray=string[];
3
4  del="";
5  del_1=" ";
6
7  emptyStr=join(emptyArray, del_1);
8          print emptyStr;
9
10 emptyDel=join(myArray, del);
11         print emptyDel;
12
13 bothEmpty=join(emptyArray, del);
14         print bothEmpty;
15
16 results=join(myArray, del_1);
17
18 print results;
19
20 return results;
```



If delimiter is empty, then the function returns all the array entries concatenated. **Example:** the following will set the `strArray1` value to "123": `strArray1=string[]{"1","2","3",};` AND `strArray2=join {strArray1,""};`

---

max(arrayIdentifier)

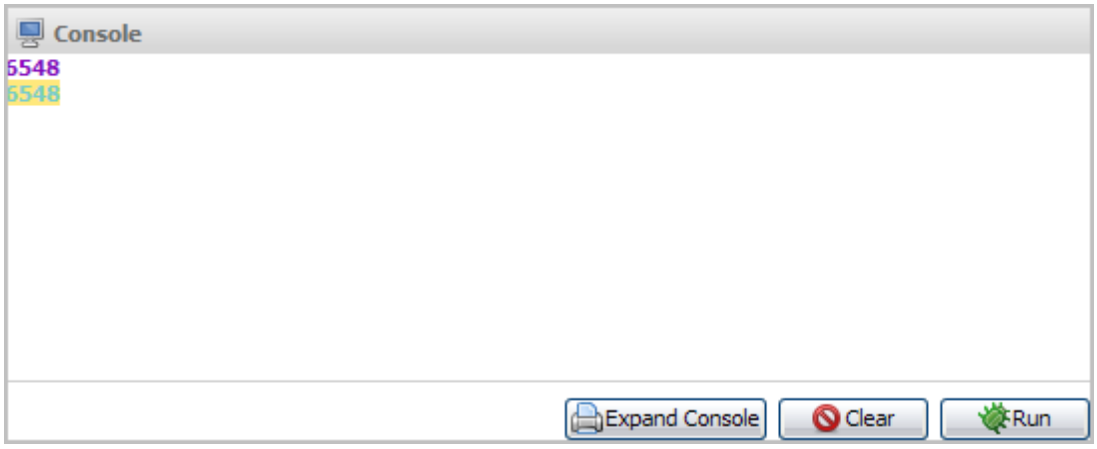
Returns the largest element of an integer or float array. This function can only be used with 1-D arrays.

#	Parameter	Data Type	Description
1	arrayIdentifier	integer[], float[]	This is the given array you will use to find the max value.

Return Type: Returns and Integer or Float depending on the type of array.

Example of max(arrayIdentifier)

```
1 myArray=integer[]{123,516,2168,6548,15};
2
3 result=max(myArray);
4
5 print result;
6
7 return result;
```



---

min(arrayIdentifier)

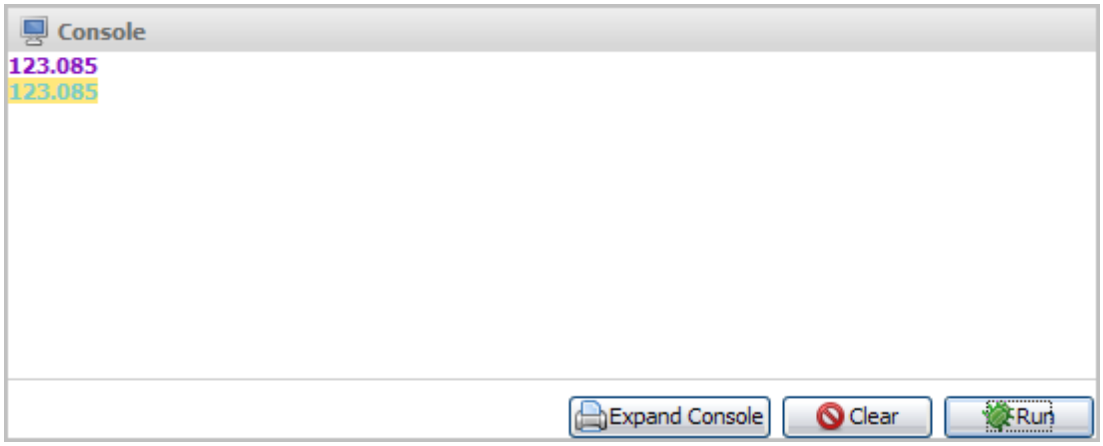
Returns the smallest element of an integer or float array. `Min()` can only be used with a 1-D array.

#	Parameter	Data Type	Description
1	arrayIdentifier	integer[], float[]	This is the given array you will use to find the max value.

Return Type: Returns and Integer or Float depending on the type of array.

Example of min(arrayIdentifier)

```
1 myArray = float[]{123.15,123.54,123.085,123.09};
2
3 result = min(myArray);
4
5 print result;
6 return result;
```



---

range(x)

Declares an integer array with a specified size and initializes it to its index value.

#	Parameter	Data Type	Description
1	x	Integer	Denotes the size of the integer array.

Return Type: Integer[]

This function is case-sensitive.

Example of range(x)

We will populate a 1-D array below. Since range declare an array of indexes, we can loop through the variable that range is equal to in order to populate a new array. Using range allows you to loop through the index values and use those to pull information out of the old arrays and input those values into the new array.

```
//Populate a 1-D array by looping through a 2-D array.

//Declare a 2-D integer array.
myArray = integer[][]{{0,12,3},{1,52,35},{53,18},{7,51,91}};
        print myArray;
//Declare your empty 1-D array.
totals = integer[];
//Create a variable to represent the index.
index = 0;
```



If x is 0, then the function returns an integer[0]. For example, `intArray=range(2)` returns an integer array size of 2 where `intArray[0]=0` and `intArray[1]=1`.

---

recordset()

The `recordSet` data type is a collection of dictionaries. It can also be used as a function in conjunction with BMQL.

Example of recordset()

```
rs = recordset();

if(...){
        rs = bmql(query1);
}
else {
        rs = bmql(query2);
}
return rs;
```



There is currently no way to return a recordSet result in a Util library function.

---

remove(arrayIdentifier, removePos)

The function removes an element from an existing array based on a given index.

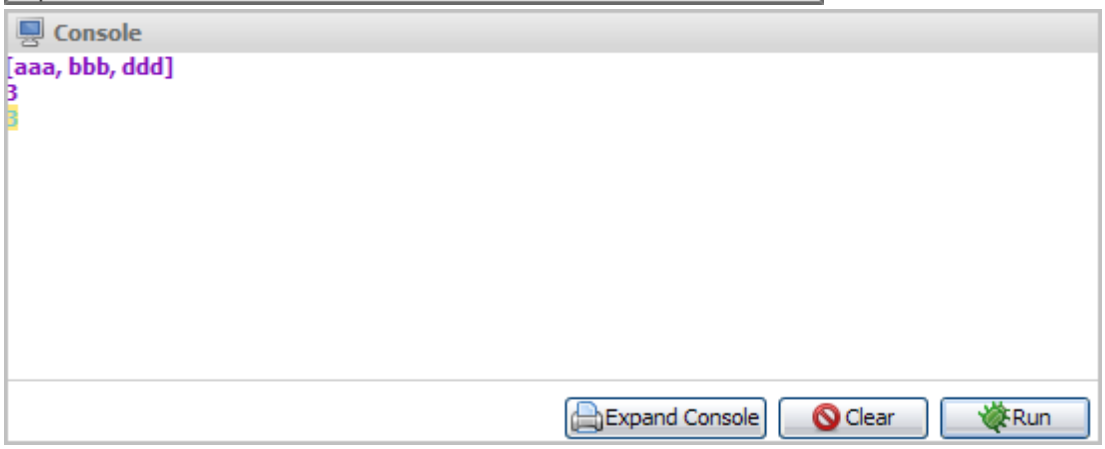
#	Parameter	Data Type	Desciption
1	arrayIdentifier	string[], integer[], float[]	This is the given array from which you will remove an element.
2	removePos	Integer	This represents the index of the element you will be removing.

Return Type: Integer *(denotes the size of the new array)*

Only integers are allowed to be passed in for the index of the element to be removed.

Example of remove(arrayIdentifier, removePos)

```
1 myArray = string[]{"aaa", "bbb", "ccc", "ddd"};
2
3 results = remove(myArray, 2);
4
5 print myArray;
6 print results;
7 return results;
```



You can see the new array that is being returned once the element at `index[2]` is removed.

If the array is empty, it will fail.

---

reverse(arrayIdentifier)

This function is used to reverse all elements in the array. `Reverse()` can only be used with 1-D arrays.

#	Parameter	Data Type	Description
1	arrayIdentifier	string[], integer[], float[]	This is the given array.

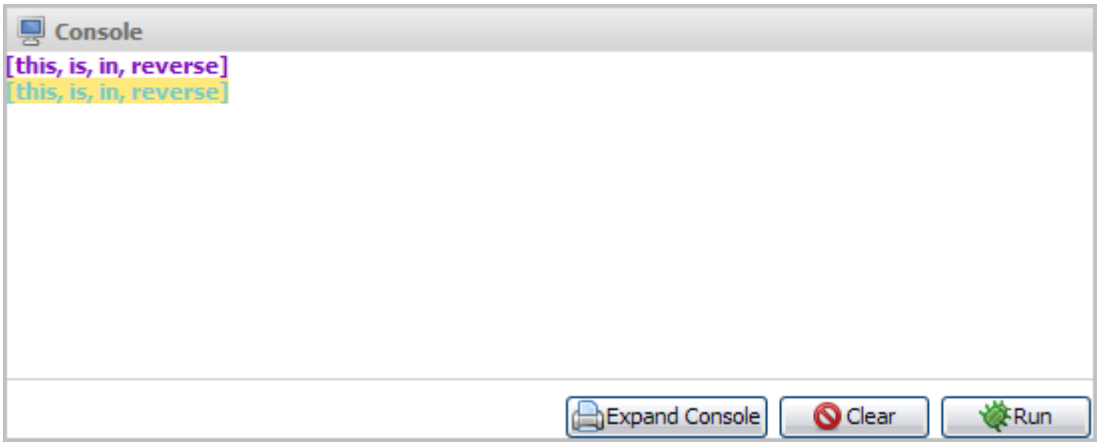
Return Type: Returns the array with all of the elements in reverse order.

Example of reverse(arrayIdentifier)

```
myArray = string[]{"reverse", "in", "is", "this"};

result = reverse(myArray);

print result;
return result;
```



---

sizeofarray(arrayIdentifier)

This function returns the length of the array for a 1-D array and the number of rows for a 2-D array.

#	Parameter	Data Type	Description
1	arrayIdentifier	string[], integer[], float[], string[][], integer[][], float[][]	This represents the given array.

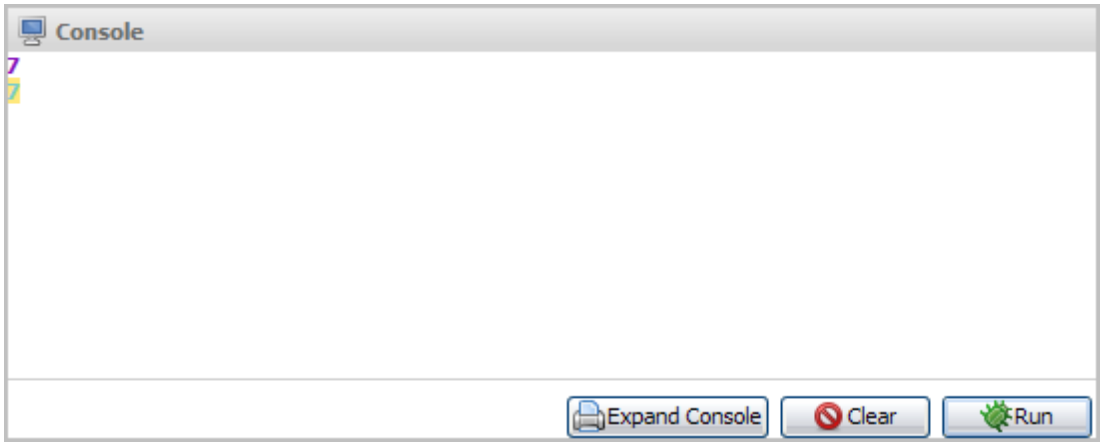
Return Type: Integer

1-D Array: sizeofarray(arrayidentifier)

```
myArray = string[]{"how", "many", "elements", "are", "in", "this", "array"};

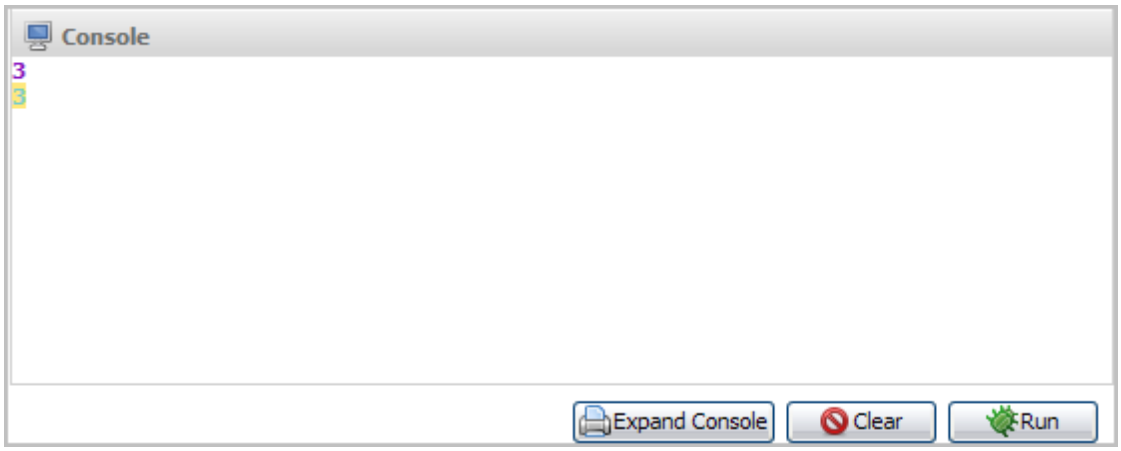
result = sizeofarray(myArray);

print result;
return result;
```



2-D Array: sizeofarray(arrayIdentifier)

```
1 myArr = integer[][]{{123,2315},{123,65484},{321,6628}};
2
3 res = sizeofarray(myArr);
4
5 print res;
6 return res;
```



While there are 6 separate values in this array, there are only three elements. Remember that `sizeOfArray` for a 2-D array will return the number of rows.

---

sort(arrayID, [sortOrder], [sortType])

This function sorts array elements based on your standard sort method. You can sort in ascending or descending order. `Sort()` can only be used with 1-D arrays.

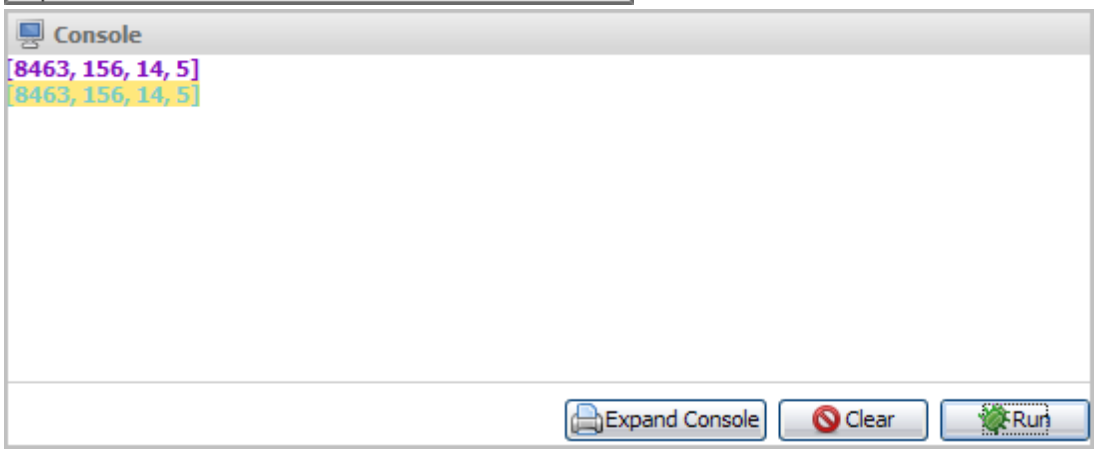
#	Parameter	Data Type	Description
1	arrayID	string[], integer [], float[]	This is the given array that you will sort.
2	[sortOrder]	String	Takes in the string values "desc" or "asc". If it is not specified, the default is "asc" or ascending. It also cannot be a variable containing the string variable. This throws an exception.
3	[sortType]	String	Takes in the string values as "text" or "numeric"

Return Type: String Array *(newly sorted)*

Example of sort(arrayID, [sortOrder], [sortType])

```
myArray = integer[]{5,156,14,8463};
newArray = sort(nyArray, "desc");

print newArray;
return newArray;
```



What we will find returned in the debugger are the values of this array rearranged into descending order by value.

---

split(str, separator)

This string function returns a 1-D string array. The function splits a string with respect to a specific separator and populates it into a string array. It also trims all of the blank spaces from results strings.

This function is case-sensitive.

#	Parameter	Data Type	Description
1	str	String	This is the given input string.
2	separator	String	Represents a variety of separators, however they all must be enclosed in double-quotes, including spaces.

Return Type: String[] *(1-D string array)*

If str is an empty string, then the function returns an empty array size of 1. If both str and separator are empty, then the function returns an empty array size of 1.

Example of split(str, separator)

```
str = "";
str_2 = "please split this string";
sep = "";
sep_1 = "";

emptyStr = split(str, sep_1);
        print emptyStr;
emptySept = split(str_2, sep);
        print emptySept;
bothEmpty = split(str, sep);
        print bothEmpty;

results = split(str_2, sep_1);

print results;
return results;
```



If separator is empty, then every character in passed string is passed. **Example:** `strArray = split ("a.b.c", "")`, will return an array with every character split, that is it will be `["a", ".", "b", ".", "c"]` with a size of 5.

RELATED TOPICS

1-D ARRAYS

OVERVIEW

The following functions initialize arrays without specifying the size. At a later time, when the values are assigned in different array indexes, the system expands the array into the appropriate size automatically.

ADMINISTRATION

String[n]

#	Parameter	Data Type	Description
1	string[]	String Array	Used to declare a string array.
2	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.

Return Type: String Array



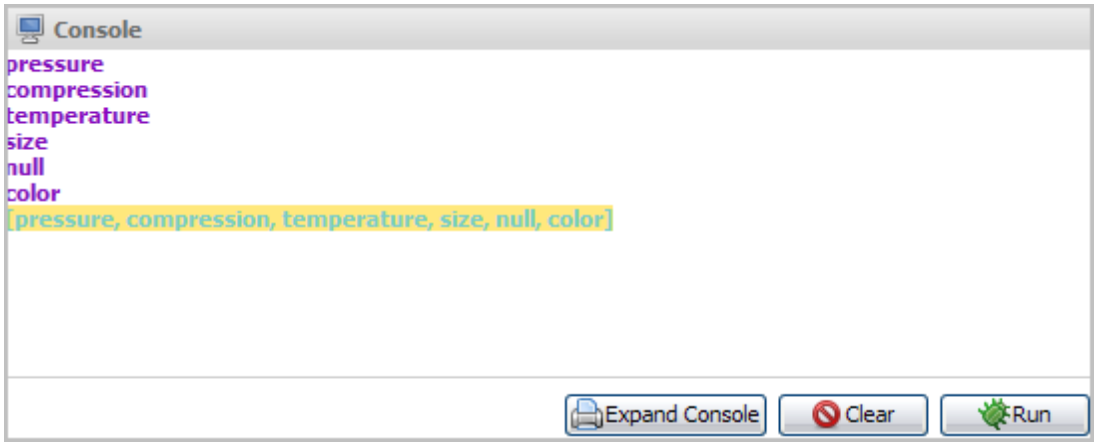
The value in each field of a String array after array declaration (e.g. `arr = string[10]`), is equal to null if the array fields are not initialized.



Negative numbers (including NaN, which is equal to -999999) specified for an array size throw a runtime exception. e.g. `arr=string[-9]`

Example of string[n]

```
1  myArray = string[]{"pressure","compression","temperature"};
2
3  print myArray[0]; //1st element in the array, "pressure"
4  print myArray[1]; //2nd element in the array, "compression"
5  print myArray[2]; //3rd element in the array, "temperature"
6
7  //Let's add a few elements
8  myArray[3] = "size"; //Adds an element to the array
9  myArray[5] = "color"; //Adds an element to the array
10
11 print myArray[3];
12 print myArray[4]; //Array field not initialized
13 print myArray[5];
14
15 return myArray;
```



These arrays are dynamic in nature and the length can be increased. E.g. `strArray=string[2];` creates a string array length of 2 and `strArray[4]="a";` increases the length of the array to 5.

Integer[n]

Just as you do for string arrays, in order to initialize an array of a particular size, use the function integer[n].

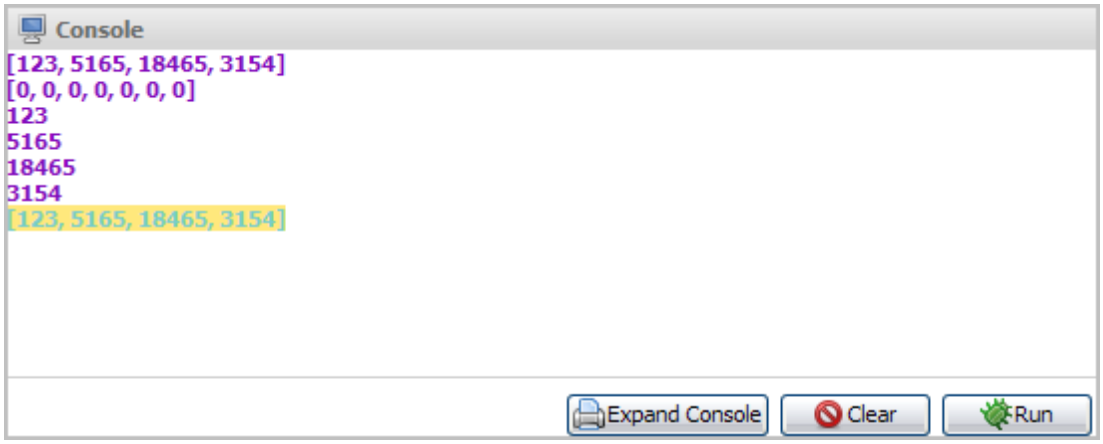
#	Parameter	Data Type	Description
1	integer[]	Integer Array	Used to declare an integer array.
2	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.

Return Type: Integer Array

Example of integer[n]

```
1  myArray = integer[]{123,5165,18465,3154};
2          print myArray;
3
4  //Define an array size without initializing array fields
5  yourArray = integer[7];
6          print yourArray;
7
8  //Print indexes
9          print myArray[0];
10         print myArray[1];
11         print myArray[2];
12         print myArray[3];
13
14 return myArray;
```

In contrast to `string[]`, if you define an array size without initializing array fields, you will see a 0 instead of null.



For Integer Array, the value in each array field, after the array declaration (e.g. `arr=integer[10];`), is equal to 0 if the array fields are not initialized.

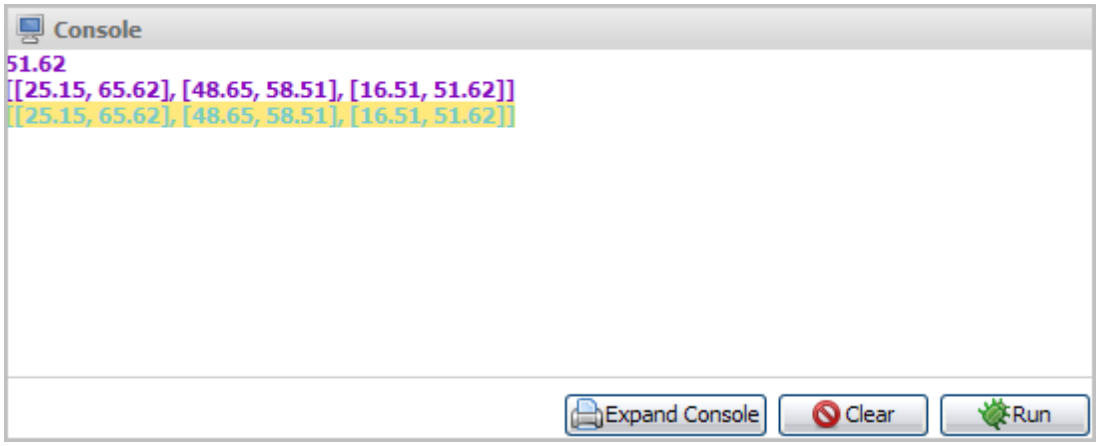
Float[n]

#	Parameter	Data Type	Description
1	float[n]	Float Array	Used to declare a float array.
2	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.

Return Type: Float Array

Example of float[n]

```
1 myArray = float[]{12.3,351.21,64.81,261.5156};
2
3 print myArray;
4
5 return myArray;
```

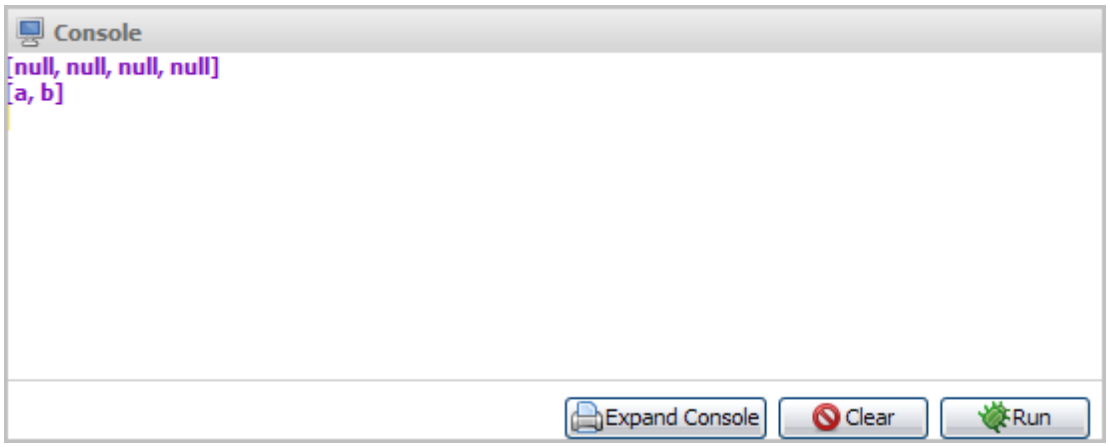


Uninitialized Arrays

In order to show you the different ways to declare arrays, we will display two separate arrays below. `myArray` will show you what to expect if you define an array size without initializing array fields. `yourArray` will show you what is returned when you define the size of an array as well as initialize the fields.

```
1  myArray = string[4];
2
3  yourArray = string[2];
4  yourArray[0] = "a";
5  yourArray[1] = "b";
6
7  print myArray;
8  print yourArray;
9
10 return "";
```

As you can see, `myArray` returned a string with four null values because array fields were not initialized after the array was declared.



NOTES

The maximum number of columns allowed in a 1-D array is 1000.

RELATED TOPICS

2-D ARRAYS

OVERVIEW

The following functions initialize arrays without specifying the size. At a later time, when the values are assigned in different array indexes, the system expands the array into the appropriate size automatically.

ADMINISTRATION

String[n][n]

#	Parameter	Data Type	Description
1	string[][]	String Array	Used to declare a 2-D string array.
2	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.
3	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.

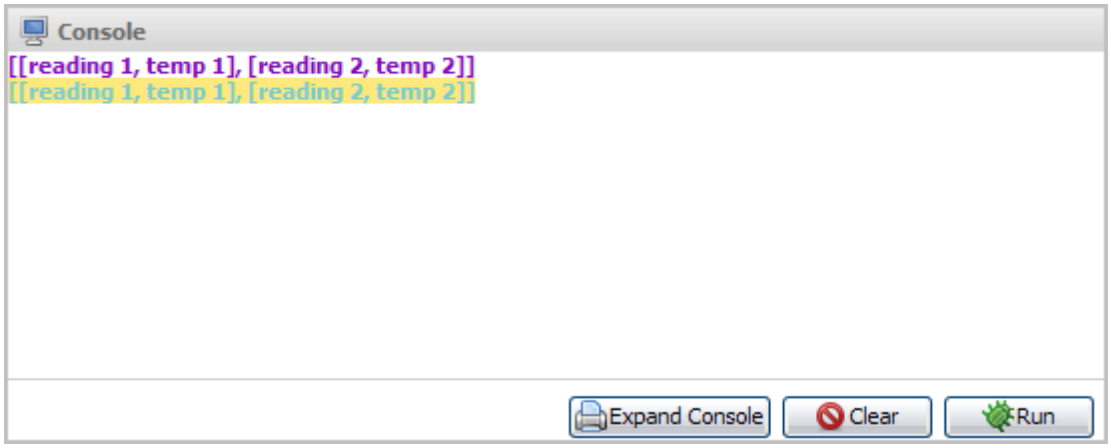
Return Type: String[][]



For String Arrays, the value in each array field, after the array initialization (e.g. `arr=string[2][3])`, is equal to null if the array fields are not initialized.

Example of string[n][n]

```
1 myArray = string[][]{{"reading 1", "temp 1"},
2                      {"reading 2", "temp 2"}};
3
4 print myArray;
5 return myArray;
```



Integer[n][n]

These arrays are dynamic in nature and the length can be increased.

#	Parameter	Data Type	Description
1	integer[][]	Integer Array	Used to declare a 2-D array.
2	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.
3	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.

Return Type: Integer[][]

For Integer Arrays, the value in each array field, after the array initialization (e.g. `arr=string[2][3])`, is equal to null if the array fields are not initialized.

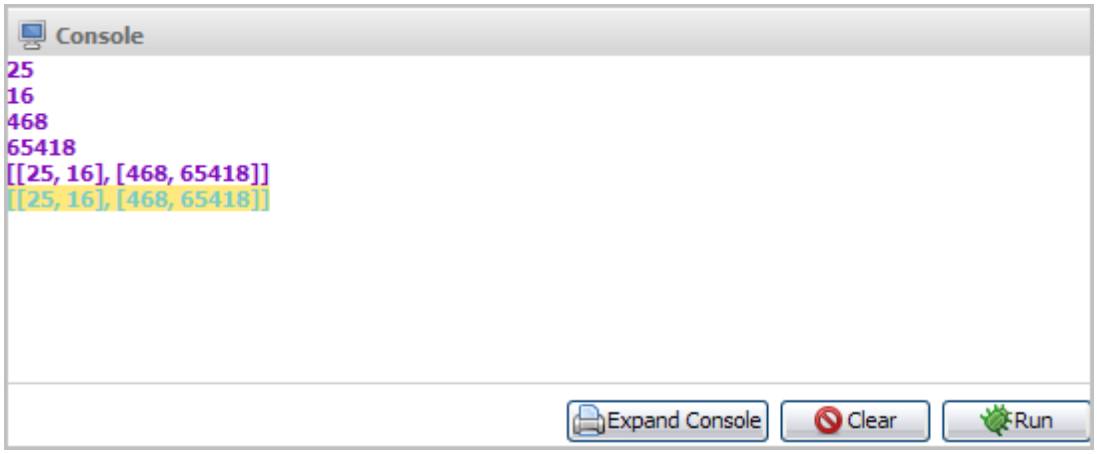
Example of integer[n][n]

```
myArray = integer[][]{{25,16},{468,65418}};

print myArray[0][0];
print myArray[0][1];
print myArray[1][0];
print myArray[1][1];

print myArray;
return myArray;
```

Notice the multiple print statements being used in the code above. We are printing a specific element index and value index. Since there are two elements, the element indexes are 0 and 1 respectively. Each element has two values, so the value indexes are also 0 and 1. The syntax used in the code will print each value in the console.



`intArray = integer[2][2];` creates an integer array with two rows and two columns.

`intArray[2][3]` automatically extends the length of the array to include an extra column.

Float[n][n]

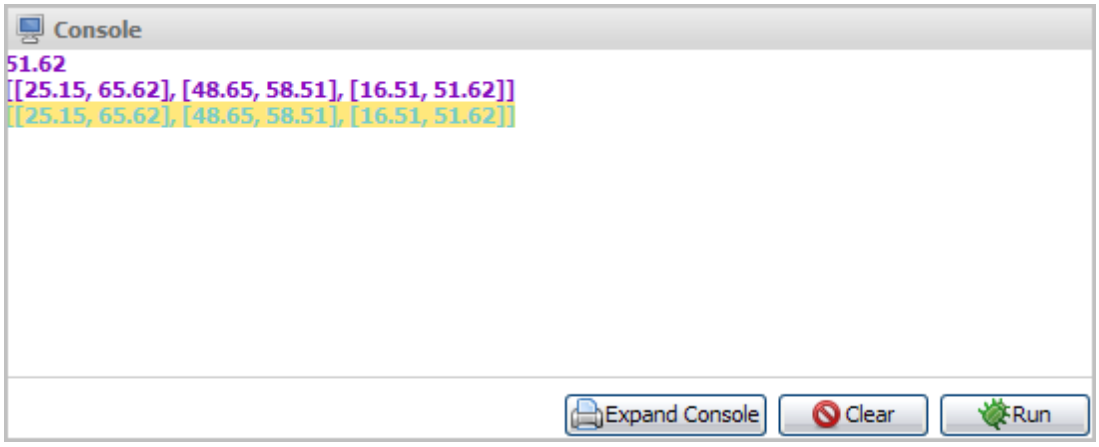
#	Parameter	Data Type	Description
1	float[][]	Float Array	Used to declare a 2-D float array.
2	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.
3	[n]	Integer	**Optional:** Use number "n" to declare an array of a specific size.

Return Type: Float[][]

For float array, the array field value after array declaration is equal to 0.0 if the array fields are not initialized.

Example of float[n][n]

```
1 floatArray = float[][]{{25.15,65.62},{48.65,58.51},{16.51,51.62}};
2
3 print floatArray[2][1];
4 print floatArray;
5
6 return floatArray;
```



These arrays are dynamic in nature and the length can be increased.

NOTES



The maximum number of rows allowed in a 2-D array is 1000 and columns is 50

RELATED TOPICS

CONDITIONAL FUNCTIONS

OVERVIEW

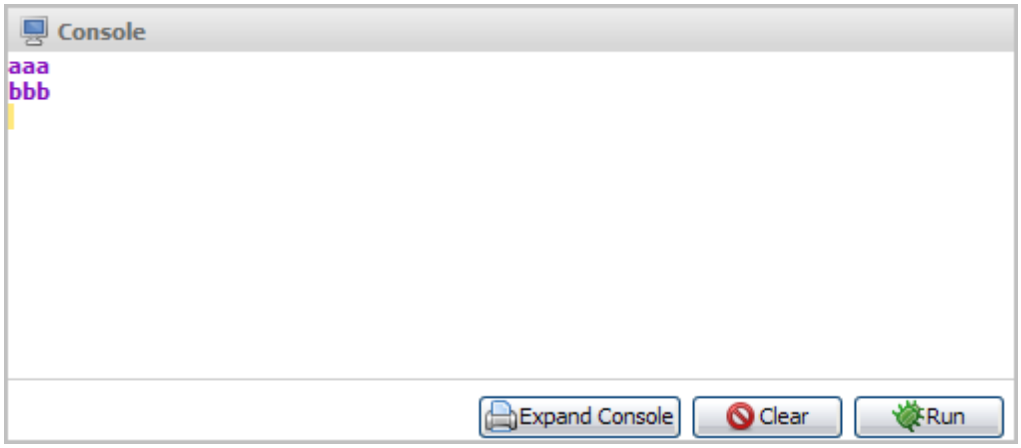
Conditional functions are commands that run only when specified criteria is met. Each conditional statement requires a Boolean expression that may or may not be visible to you. To put it more plainly, a conditional function must evaluate to true or false.

ADMINISTRATION

Break

Using this function will break a `for...loop` at the element that you define. In this example, we used a 1-D string array and asked it to stop when it reached the element "ccc".

```
1  myArray=string[]{"aaa", "bbb", "ccc", "ddd"};
2
3  for each in myArray{
4          if (each=="ccc"){
5          break;
6
7  }else{
8          print (each);
9          }
10 }
11
12 return "";
```



Continue

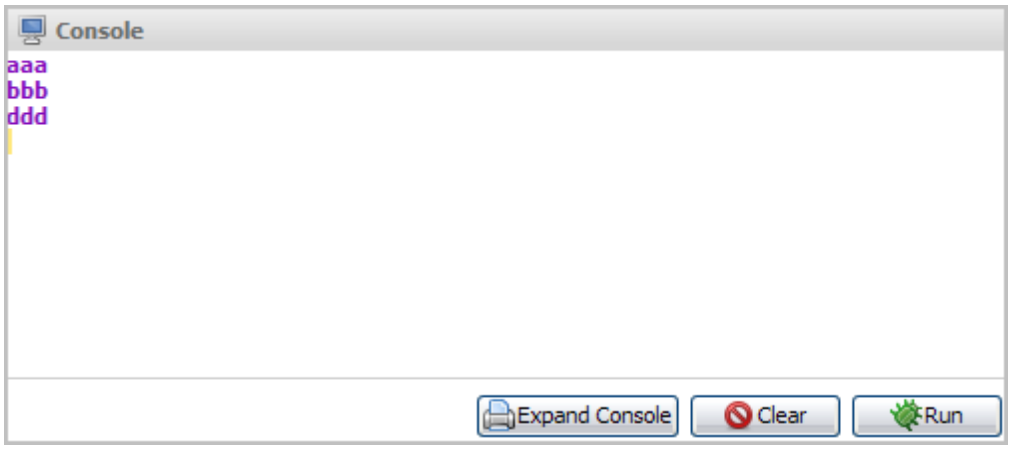
The `continue` action calls out an element from an array and essentially skips it, but continues looping through the rest of the array. Unlike the break function, the loop continues after it skips over the element.

```
myArray=string[]{"aaa", "bbb", "ccc", "ddd"};

for each in myArray{
        if (each=="ccc"){
        continue;

}else{
        print (each);
        }
}

return "";
```



For...Loop

A `for...loop` is a popular function that is used in many programming languages. A `for...loop` statement is meant to loop through a block of code until a specific condition is met.

The basic syntax is:

```
for(variable) in (system_array) {
statement(s);
}
```

The variable represents the element, but you can label this however you want. You can call it 'each' or 'element' or even your name if you wanted to. Let's think of a table of values that we will name `Table_Array`:

	Column 1	Column 2	Column 3
Row 1	1	2	3

Now, let's use the same syntax as above:

```
for(row element) in Table_Array{
print row element;
}
```

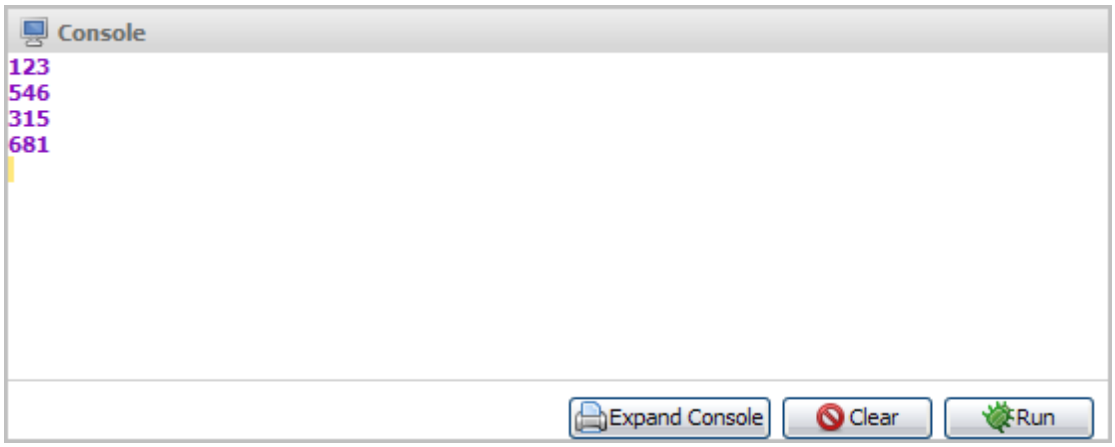
So, what we are really saying is: "For Row 1 in `Table_Array`, print each element in the row".

Example of for...loop in 1-D Array

In the following example, we are setting up a 1-D integer array. What this code is saying is, "for whatever array element is in `myArray`, print that array element".

```
1 myArray=integer[]{123,546,315,681};
2
3 for arrElement in myArray {
4         Print arrElement;
5 }
6
7 return "";
```

So, the code is going to loop through the integer array we have defined and print every element that is in this 1-D array.



Example of for...loop in 2-D Array

Now, let's say we have a table that has more than one row.

	Column 1	Column 2
Row 1	0	3
Row 2	1	4
Row 3	2	5

You will have to use a loop within a loop in order to return the row values and the column values.

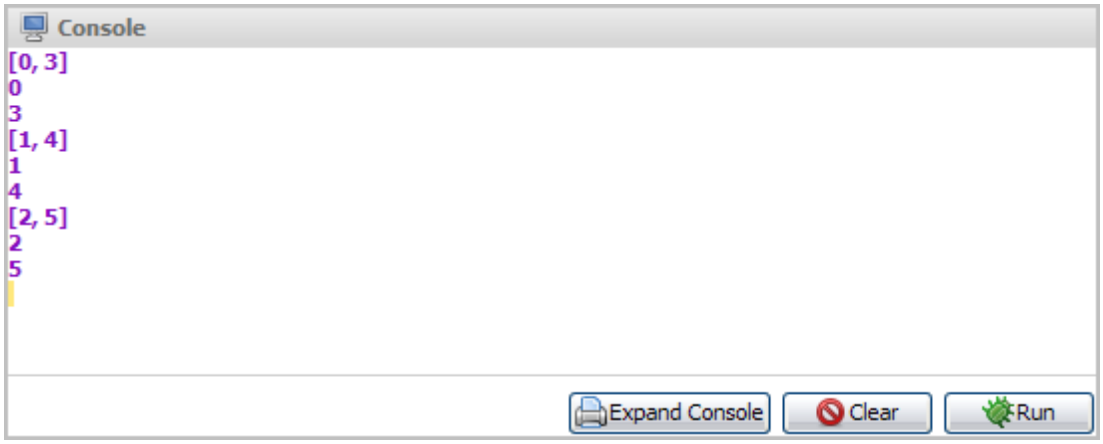
```
myArray=integer[][]{{0,3},{1,4},{2,5}};


for row in myArray{
        print row;
        for col in row{
                print col;

        }
}

return "";
```

So, what we will see is that for each row, it will print that row {Row 1 will return (0,3)}, after it loops through and pulls the value for the row, it's going to loop through the row and pull the values for the columns, returning to you the individual values of 0 and 3.



If...Statements

What this is saying in plain language is: "If (condition) is True, then perform the statement". If that condition is False, then no action is taken. This is a very basic example, saying that if the variable string is a number, then we want to convert it to a float.

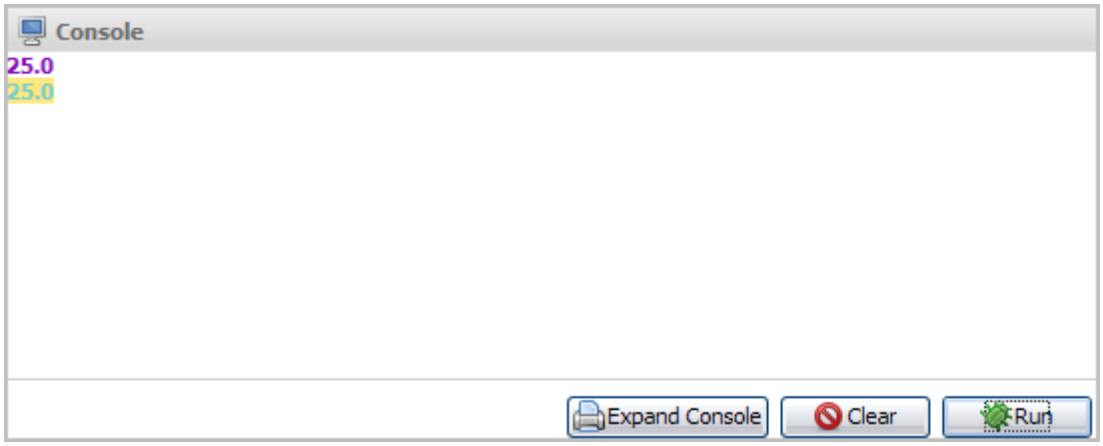
```
var="25";

if(isnumber(var)){
        results=atof(var);
}

print results;

return results;
```

In this case, "25" is a number, so we used `atof()` to convert the string to a Float.



If...Else...Statements

An `if...else` statement is essentially the same as an `if...` statement, except that instead of doing nothing if the condition isn't met, a different statement is run.

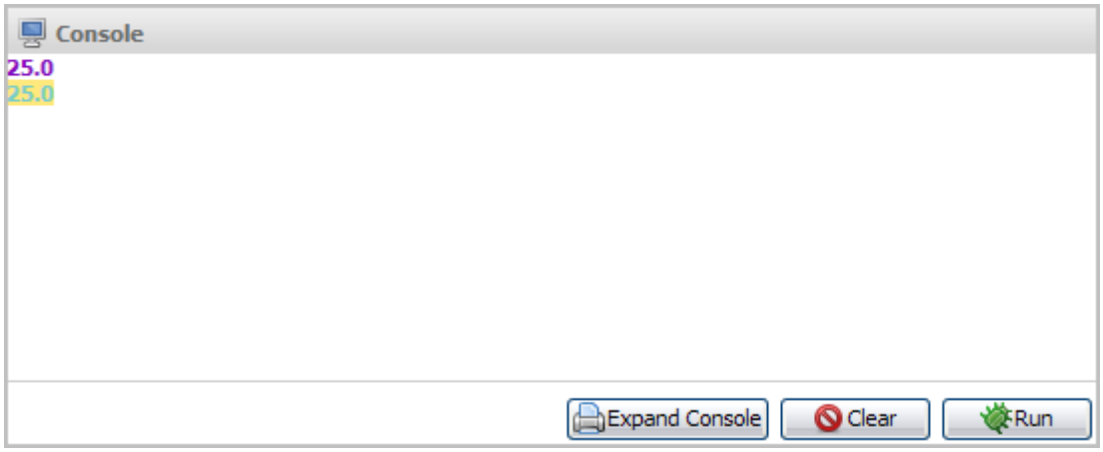
The basic syntax is:

```
if (condition){
statement;
} else {
statement;
}
```

So, it's saying "Do something if the first condition is met, if not, do this instead". You can also have multiple else statements nested within one another.

```
1  var="25";
2
3  if(isnumber(var)){
4          results=atof(var);
5          }else{
6                  print "NaN";
7  }
8
9  print Results;
10
11 return results;
```

We are saying, "if the var = 25" is a number, then convert the string to a Float. If it's not a number, print "NaN".



RELATED TOPICS

DATE FUNCTIONS

OVERVIEW

Date functions are often used when evaluating and calculating quotes in Commerce. As an example, they are used to set quote expiration dates. Dates are always strings in Commerce, but can be converted to dates using the functions included in this article.

ADMINISTRATION

adddays(date, num_of_days)

This function returns the date which is obtained after adding X number of days to a particular date. **Example Use Case:** Often used in Commerce to set the time a quote is valid, or when it expires.

#	Parameter	Data Type	Description
1	date	Date	Represents the given date.
2	num_of_days	Integer	How many days you want to add to the given date.

Return Type: Date

Example of `adddays(date, num_of_days)` *We will use multiple functions that will accomplish the following:*

- Add 60 days to the current date.
- Convert that date to a string format.
- Remove the `hours:minutes:seconds` from the return.

```
1  findDate=(datetostr(adddays(getdate(), 60)));
2          print findDate;
3
4  end=len(findDate);
5
6  finalDate=substring(findDate, 0, end-8);
7
8  print finalDate;
9
10 return finalDate;
```

---

datetostr(date)

This function converts date to a string. It can take an optional parameter that can return a formatted string. **Example Use Case:** Date fields in commerce are considered string fields so to return a date to a commerce attribute, you need to convert it to a string first.

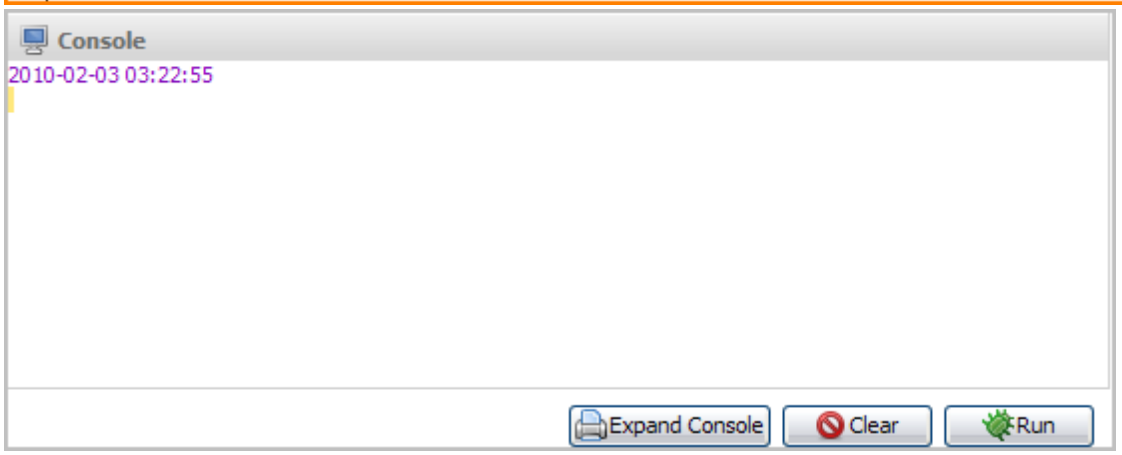
#	Parameter	Data Type	Description
1	date	Date	Mandatory date parameter.
2	[dateFormat]	String	Optional parameter that returns a formatted string.

Return Type: String

Valid Date Components	Description
yyyy	Year in long form specifying all digits
MM	Month in numeric format
dd	The day of the month in numeric form *e.g. 24*
HH	The hour as a number in 24 hour form
mm	Minutes
ss	Seconds

Example of datetostr(date, [dateFormat])

```
1 //Using enhanced datetostr function and its option parameter
2 testDate = strtojavadate("02/03/2010 03-22-55", "MM/dd/yyyy HH-mm-ss");
3 dateStr = datetostr(testDate, "yyyy-MM-dd HH:mm:ss");
4
5 print dateStr;
6
7 return "";
```



datetostr (getdate()) will return today's date in the format MM/dd/yyyy HH:mm:ss if no format is supplied

---

getdate() AND getstrdate()

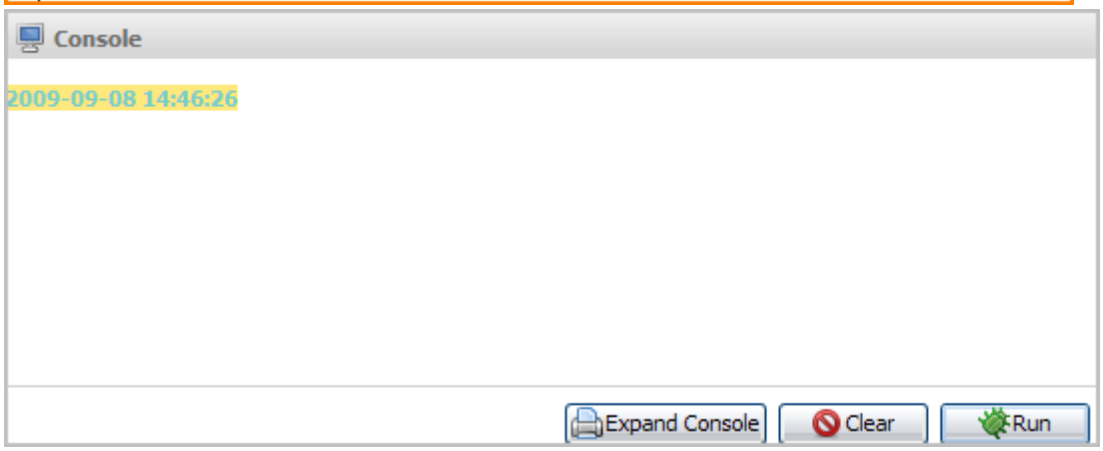
This function returns the current date/time based on the base time zone you have set-up in your application. `getdate()` takes a parameter and returns a date with or without time. **Example Use Case:** Find the current date.

#	Parameter	Data Type	Description
1	[includeTime]	Boolean	Optional parameter that will return the current system date and time. If it is not included, it will be set to TRUE by default. If this parameter is set to false, the component is set to 00:00:00 and is not displayed to the user.

Return Type: Date

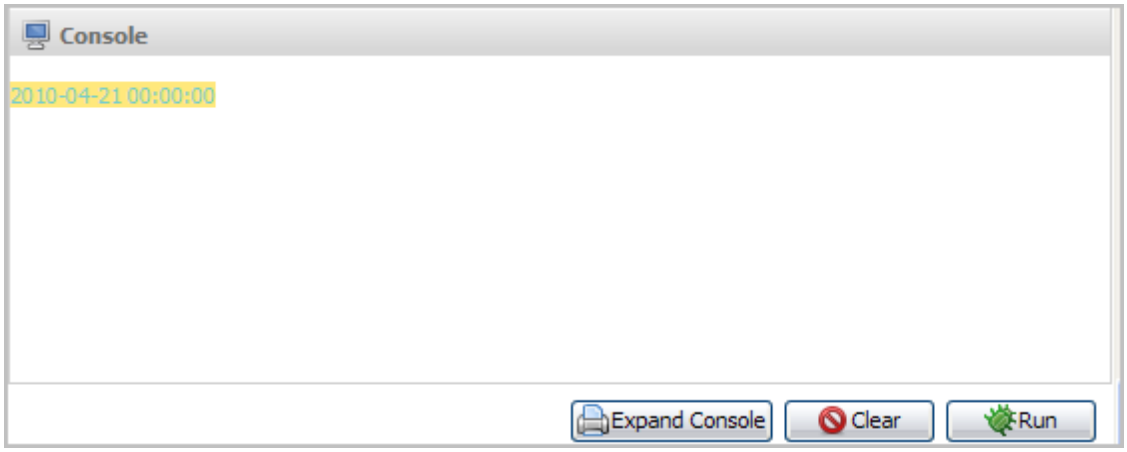
Example of getdate()

```
1 return getdate();
```



Example of getdate(false)

```
1 return getdate(false);
2
```



---

getdiffindays(date 1, date 2)

This function calculates the number of days between two different dates. **Example Use Case:** Check to see if a quote is expired.

#	Parameter	Data Type	Description
1	date 1	Date	Mandatory date parameter.
2	date 2	Date	Mandatory date parameter.

Return Type: Float

Example of getdiffindays(date 1, date 2)

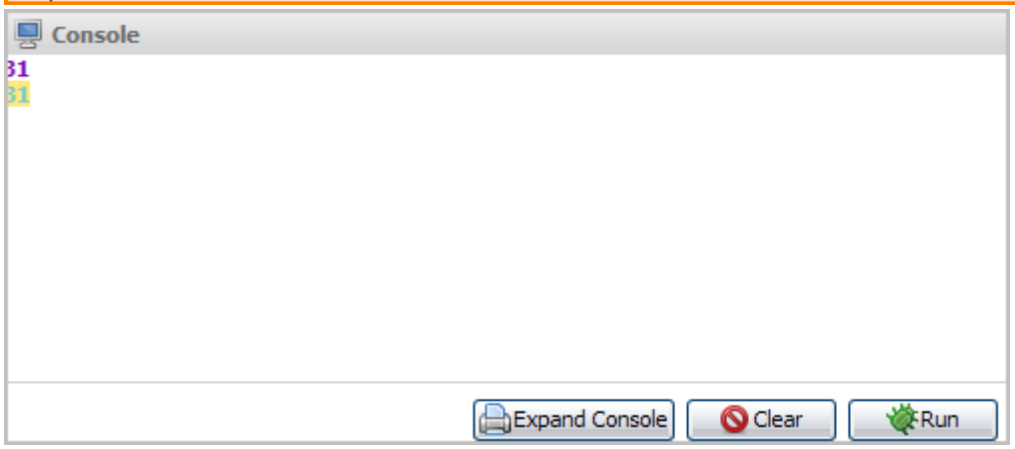
```
todayDate=getdate();
otherDate="10/09/2009";

interim=strtojavadate(otherDate, "MM/dd/yyyy");

results=getdiffindays(todayDate, interim);

print results;

return results;
```



You must use strtojavadata() to change otherDate from a string format to a date format.

The inputs must be in Date format.

---

isleap(year_num)

Using this function will determine whether the date falls within a leap year. The function will return true if they year provided as a parameter is a leap year. **Example Use Case:** Can determine if the date entered by a user is a valid date; can be used when trying to calculate the total number of days in a given year.

#	Parameter	Data Type	Description
1	year_num	Integer	Four-digit integer that represents the year in question .

Return Type: Boolean

Example of isleap(year_num)

```
dateVar=isleap(2008);

print dateVar;

return dateVar;
```

Since the year 2008 was a leap year, the function returns True.

---

isweekend(date)

This will determine whether a date falls within a weekend (Saturday or Sunday). **Example Use Case:** Can determine if the date entered by a user is a valid date.

#	Parameter	Data Type	Description
1	date	Date	Mandatory date parameter.

Return Type: Boolean

Example of isweekend(date)

```
dateVar = isweekend(getdate());

print dateVar;
return dateVar;
```

The console will return either True or False.

---

minusdays(date, num_of_days)

This function returns a date that is x days before the base date. **Example Use Case:** Can be used to set an expiration date of a quote.

#	Parameter	Data Type	Description
1	date	Date	Represents the given date.
2	num_of_days	Integer	How many days you want to subtract from the given date.

Return Type: Date

Example of minusdays(date, num_of_days)

```
todaysDate = getdate();
results = minusdays(todaysDate, -45);

return results;
```

---

strtojavadate(str, format)

This function converts a string to a date and replicates Java behavior. This function is preferred over `strtodate()`, which is a deprecated function that will no longer be supported.

#	Parameter	Data Type	Description
1	str	String	String format of a date.
2	format	Date Format	Specifies the format the date will be returned in.

Date Format	Date Type
dd	Date
MM	Month
YY	Year
HH	Hour in Day (0-23)
Kk	Hour in Date (1-24)
KK	Hour in AM/PM (0-11)
Hh	Hour in AM/PM (1-12)
mm	Minute
ss	Second

Return Type: Date

Example of strtojavadate(str, format)

```
1 todayDate = "01/01/2013";
2 results = strtojavadate(todayDate, "MM/dd/yyyy");
3
4 print results;
5 return results;
```



The format strings supported are different from the ones supported in strtodate().



If str is an empty string, then this method will thrown an exception, unlike the strtodate() method.

NOTES



Dates always need to be returned as strings to Commerce.

RELATED TOPICS

NUMERIC FUNCTIONS

OVERVIEW

Numeric functions serve many purposes throughout the BigMachines application. Many of them represent some basic mathematical and trigonometric functions.

asin(x)

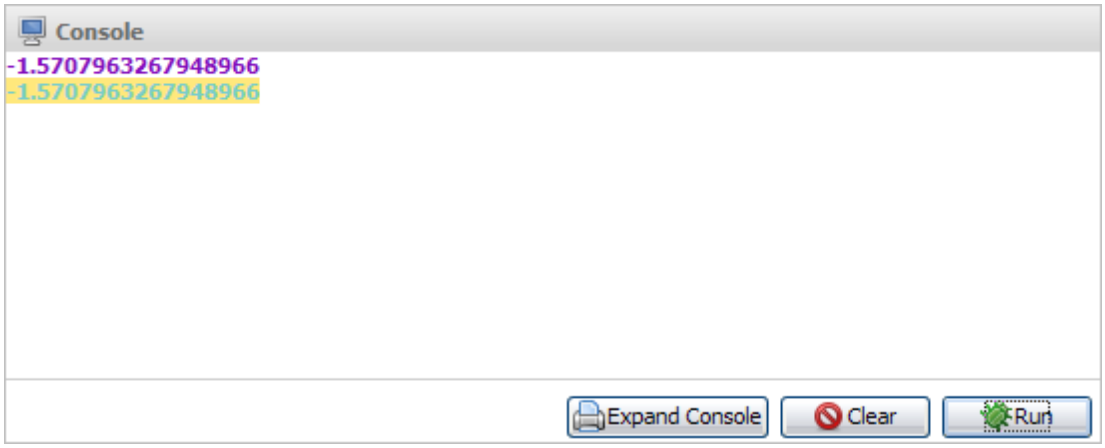
The usual notation of the inverse trig function arcsine is, `y = arcsine(x)`, which is also defined as `x = sin(y)`. This notation is the same for arccosine and arctangent.This function returns the arcisne of a number or angle, the range of which is: `-π/2 ≤ y ≤ π/2 or -90° ≤ y ≤ 90°` and the domain is `-1 ≤ x ≤ 1`. **Example Use Case:** Find the inverse function of the sine of an angle.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

Example of asin(x) *If you enter your float variable as (-1), which is the lowest number in the domain, it will return the lowest number in the range, which is -π/2.*

```
1 sizeofAngle=-1;
2
3 results=asin(sizeofAngle);
4
5 print results;
6
7 return results;
```



Float variable > 1 will return NaN

---

acos(x)

This function returns the arccosine of the number or angle in the range of 0 through π.**Example Use Case:** Find the inverse function of the cosine of an angle.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

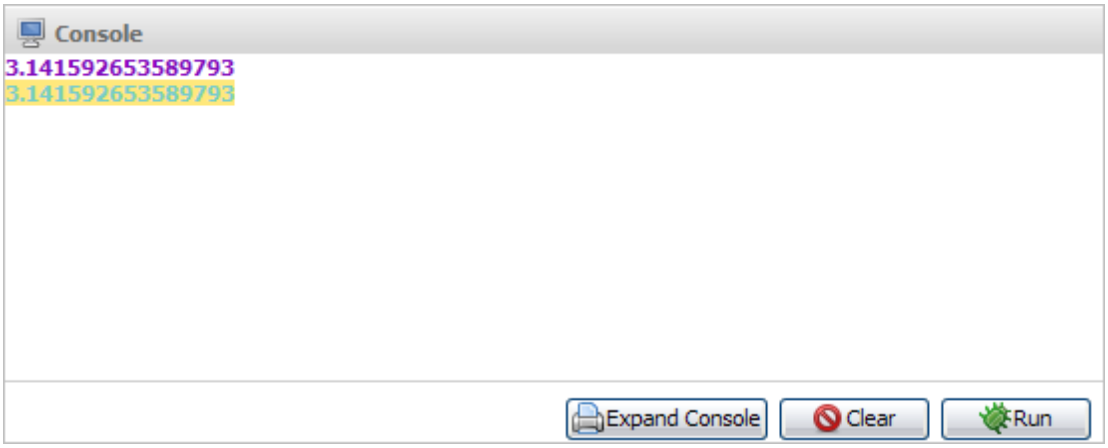
Example of acos(x)

```
sizeofAngle=-1;

results=acos(sizeofAngle);

print results;

return results;
```





Float variable > 1 returns NaN

---

atan(x)

This function returns the arctangent of the number or angle, the range of which is: -π/2 ≤ y ≤ π/2. **Example Use Case:** Find the inverse function of the tangent of an angle.

#	Parameter	Data Type	Desciption
1	x	Float	Represents a given float input parameter.

Return Type: Float

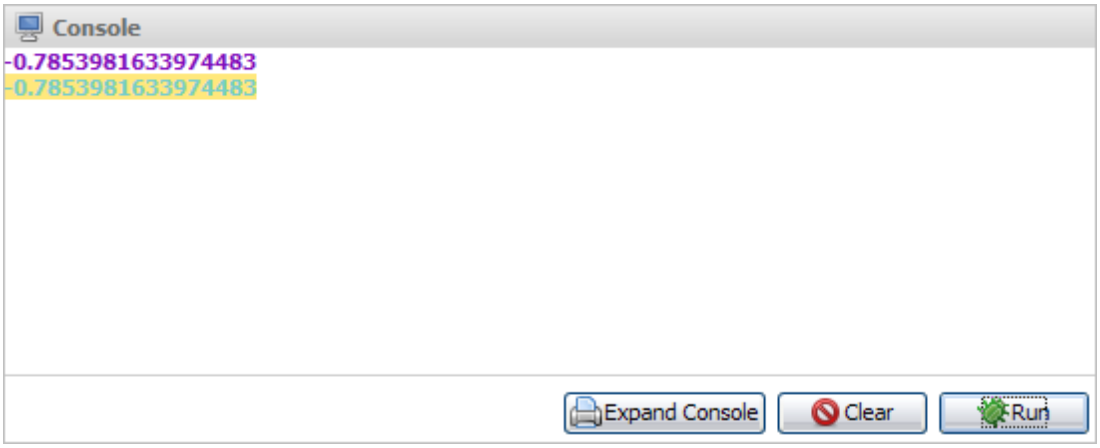
Example of atan(x)

```
sizeofAngle=-1;

results=atan(sizeofAngle);

print results;

return results;
```



---

ceil(x)

This function converts a float into the next highest whole number. **Example Use Case:** Rounding up to the next highest whole number.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

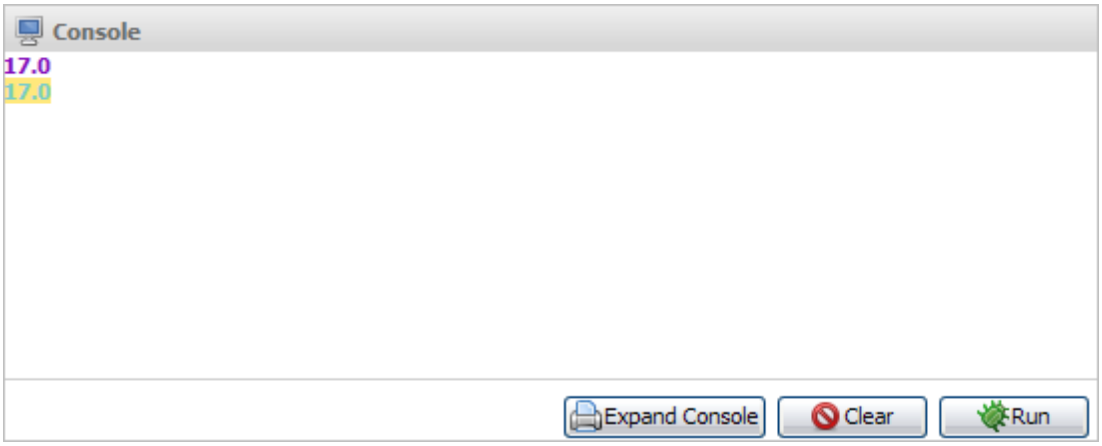
Example of ceil(x)

```
floatVar=16.54612354;

results=ceil(floatVar);

print results;

return results;
```



If -1.0 < x < 0, then the result is negative zero.

---

cos(x)

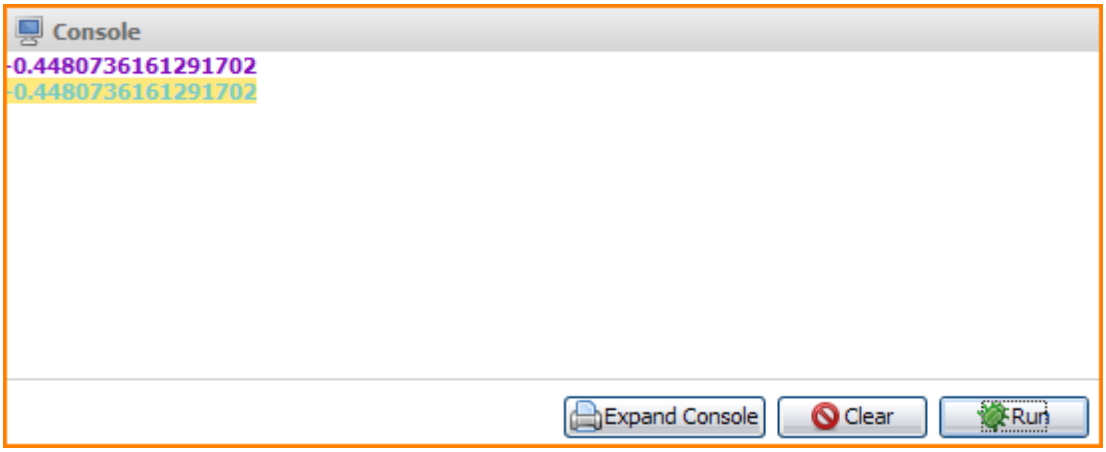
This function returns the cosine of the number/angle. **Example Use Case:** Find the ratio of the side adjacent the given angle and the hypotenuse.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

Example of cos(x)

```
1 sizeofAngle=90.0;
2
3 results=cos(sizeofAngle);
4
5 print results;
6
7 return results;
```



---

cosh(x)

This function returns the hyperbolic cosine of a number or angle.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

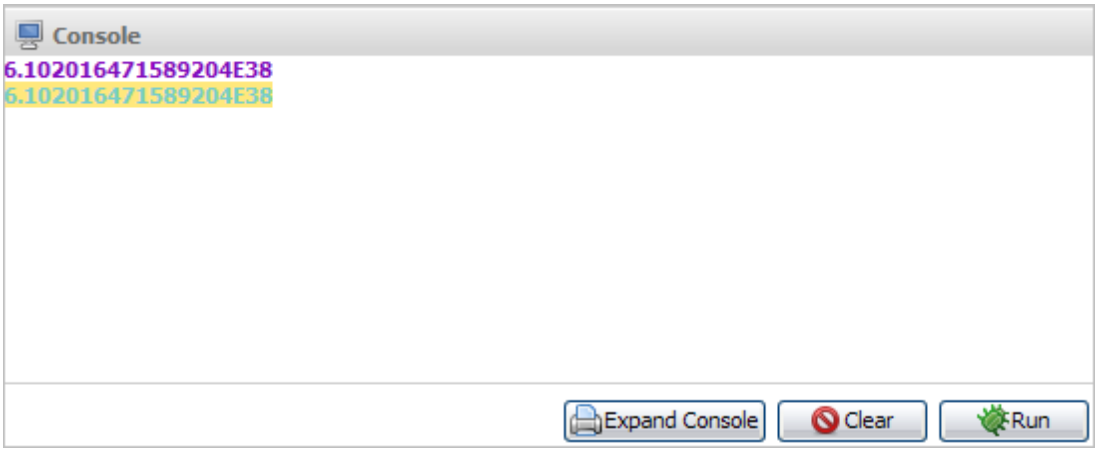
Example of cosh(x)

```
sizeofAngle=-90;

results=cosh(sizeofAngle);

print results;

return results;
```



cosh x = 1/2(ex + e-x)

---

exp(x)

This function returns ruler's number e raised to the power of the number passed through the function, also known as the exponential function. **Example Use Case:** Find whether a quantity grows or decays at a rate proportional to its current value, such as compound interest.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

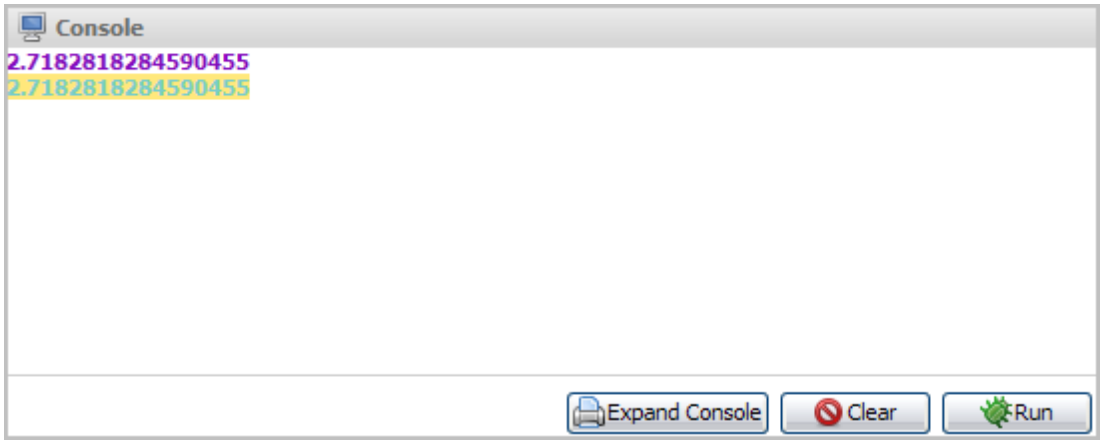
Example of exp(x)

```
floatVar=1.0;

results=exp(floatVar);

print results;

return results;
```



---

fabs(x)

This function returns the absolute value of a number. **Example Use Case:** Finding the distance of the quantity from zero.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

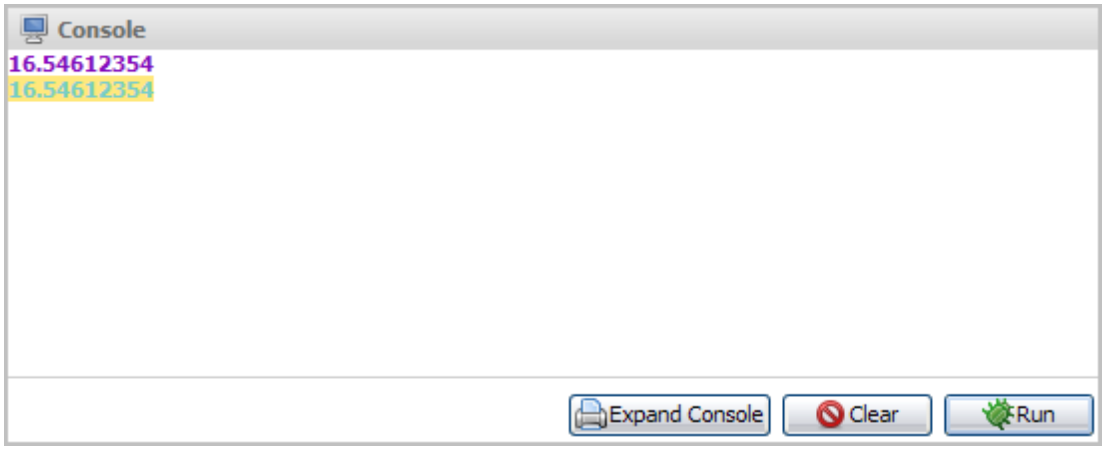
Example of fabs(x)

```
floatVar=-16.54612354;

results=fabs(floatVar);

print results;

return results;
```



fmod(x,y)

This function returns the remainder after division.

#	Parameter	Data Type	Description
1	x	Float	Represents the dividend.
2	y	Float	Represents the divisor.

Return Type: Float

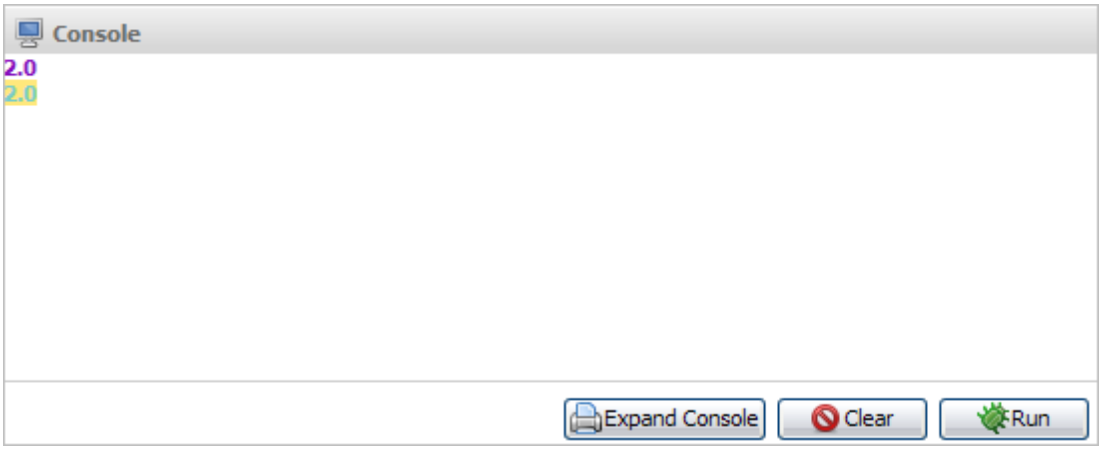
Example of fmod(x,y)

```
floatVar=17.0;
divBy=3.0;

results=fmod(floatVar, divBy);

print results;

return results;
```



If you were to use long division, the answer is 5 with a remainder of 2. If you plus this operation into your calculator, you will see the answer is 5.6667, because the remained is being divided by the divisor.

hypot(x,y)

This function returns the `sqrt(x2 + y2)` without intermediaries. **Example:** `sqrt(32 + 42) = sqrt(25) = 5.0` or `-5.0`.

#	Parameter	Data Type	Description
1	x	Float	Represents the x in sqrt (x2 + y2).
2	y	Float	Represents the y in sqrt (x2 + y2).

Return Type: Float

Example of hypot(x,y)

```
floatVar=10.0;
floatVar_2=7.0;
roundtoVar=3;

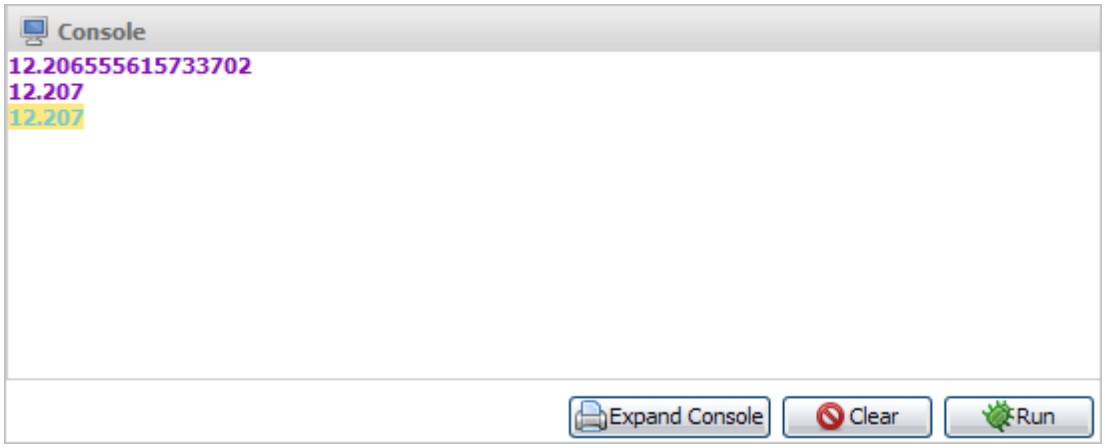
res=hypot(floatVar, floatVar_2);

print res;

results=round(res, roundtoVar);

print results;

return results;
```



As you can see, we have declared three variables. The two that were required for this function to work, as well as a third variable so we can round the results of the `hypot()` function. The result of this function return a rather large number. Using the `round()` function, we were able to round to the third decimal place.



Negative numbers can be passed as parameters to this function.

---

integer(x)

This function returns the integer portion of a float number. For example, this function would return 14 for the number 14.3345324.
Example Use Case: Returning an attribute as an integer, without rounding.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Integer

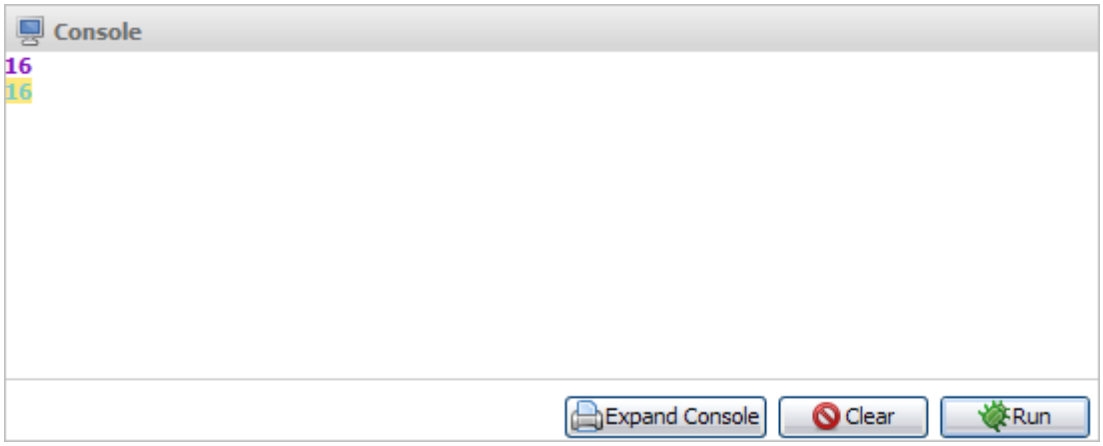
Example of integer(x)

```
floatVar=16.54612354;

results=integer(floatVar);

print results;

return results;
```



---

log(x)

This function returns the base-10 logarithm of the number.

Example: If 1000 = 103, then 3 = log10(1000). The logarithm of 1000 would be 3 because that is how many times you must multiply 10 to get 1000.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float parameter.

Return Type: Float

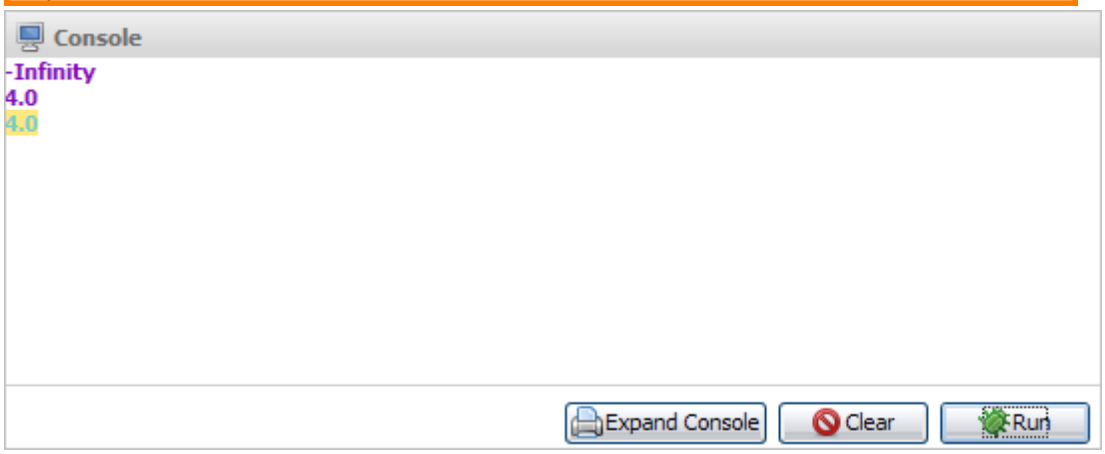
Example of log(x)

```
floatVar=0.0;
floatVar_2=10000.0;

infTest=log(floatVar);
        print infTest;

results=log(floatVar_2);
        print results;

return results;
```



If x = bx, then y = logb(x).

log(0) or log(-0) is negative infinity AND log(infinity) is infinity.

---

ln(x)

This function returns the natural logarithm (base e) of the number.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float parameter.

Return Type: Float

Example of ln(x)

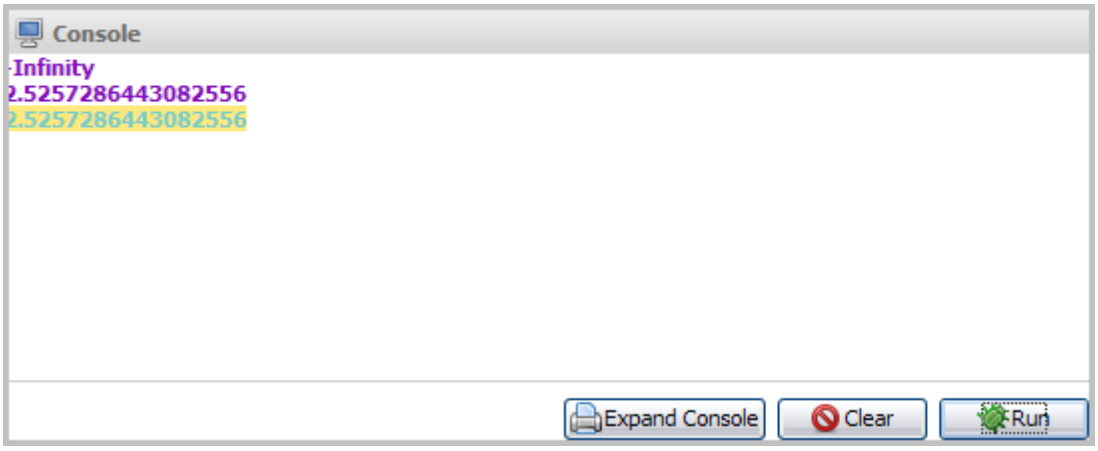
```
floatVar=12.50;
floatVar_2=0.00;

infinityRes=ln(floatVar_2);
        print infinityRes;

results=ln(floatVar);

print results;

return results;
```



If e3 = x, then ln(x) = 3.

Special Rules:
ln(1.0) =)
ln(0) or ln(-0) = infinity
ln(infinity) = infinity

---

pow(x,y)

This function returns the value of the first argument raised to the power of the second argument. **Example Use Case:** Finding the product obtained from multiplying a quantity by itself one or more times.

#	Parameter	Data Type	Description
1	x	Float	Represents the value of the input float.
2	y	Float	Represents the power to which you raise the input float.

Return Type: Float

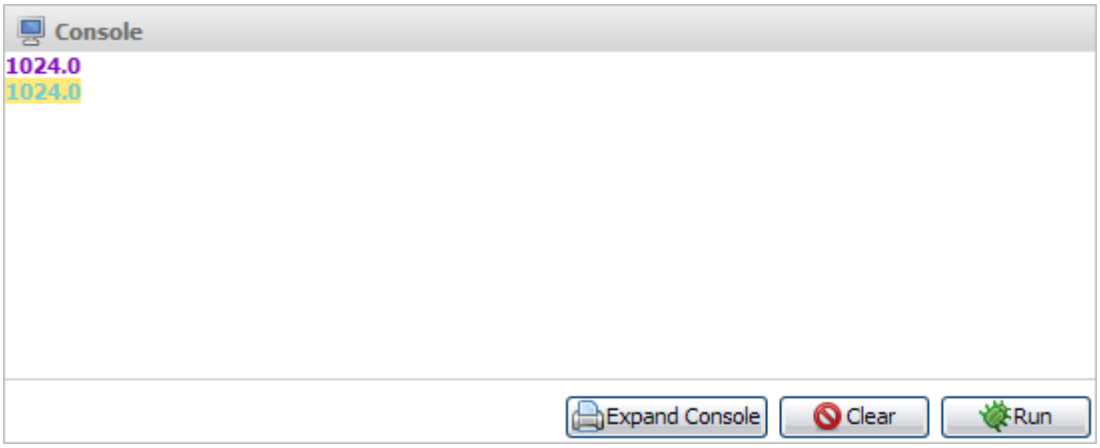
Example of pow(x)

```
floatVar=4.0;
raisetoVar=5;

results=pow(floatVar,raisetoVar);

print results;

return results;
```



---

round(x)

This function returns the rounded value of a number up to a certain decimal point. **Example Use Case:** For returning float values that represent currency. You may want to round to a certain decimal place.

#	Parameter	Data Type	Description
1	x	Float	Represents the given input float parameter.
2	[n]	Integer	Optional: Specifies the number of decimal places to round to.

Return Type: Float

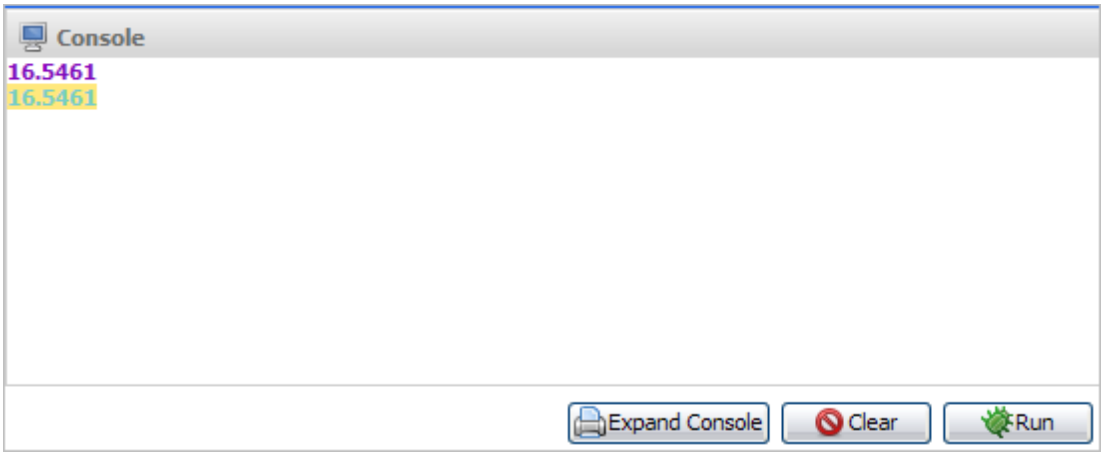
Example of round(x)

```
floatVar=16.54612354;
decVal=4;

results=round(floatVar, decVal);

print results;

return results;
```



---

sin(x)

This function returns the trigonometric sine of the number/angle. **Example Use Case:** Finding the ratio of the size opposite the given angle and the hypotenuse.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

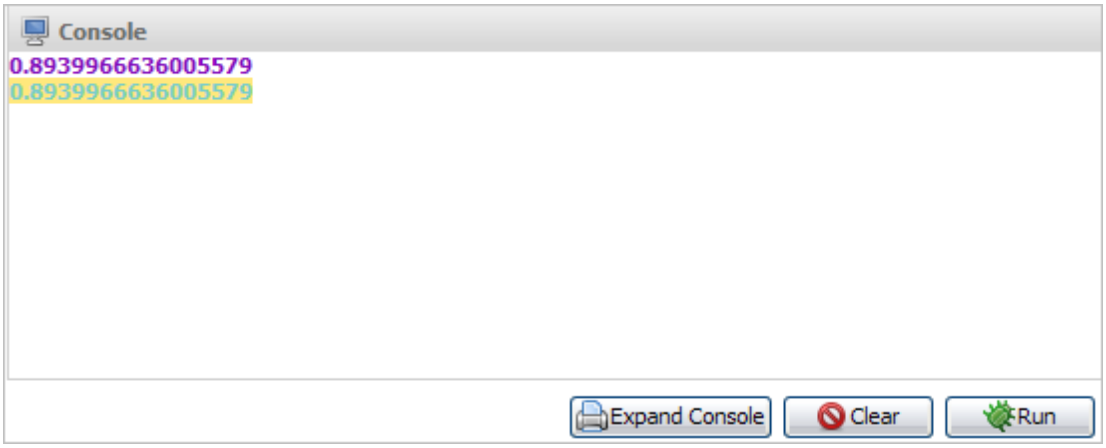
Example of sin(x)

```
sizeofAngle=90.0;

results=sin(sizeofAngle);

print results;

return results;
```



---

sinh(x)

This function returns the hyperbolic sine of a number or angle.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

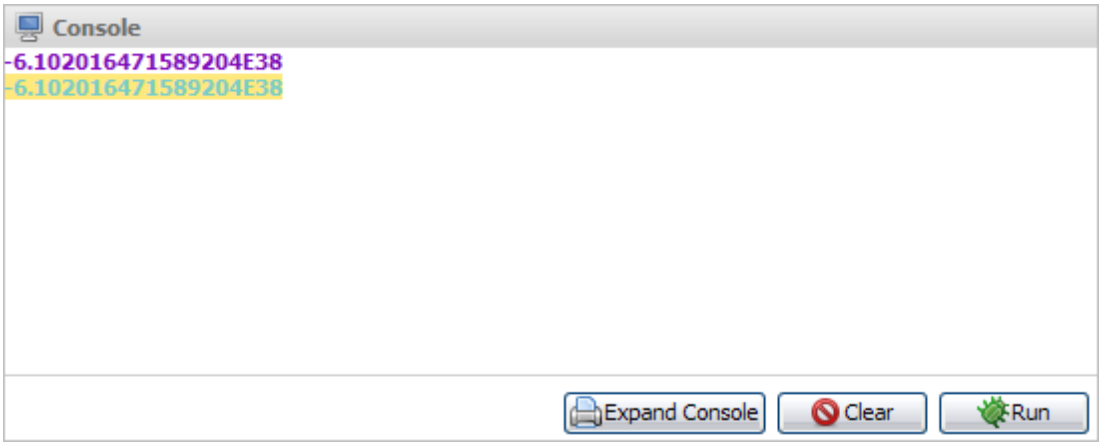
Example of sinh(x)

```
sizeofAngle=-90;

results=sinh(sizeofAngle);

print results;

return results;
```



sinh x = 1/2(ex - e-x)

---

sqrt(x)

This function returns the positive square root of a number.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

Example of sqrt(x)

```
floatVar=49.0;

results=sqrt(floatVar);

print results;

return results;
```

---

tan(x)

This function returns the tangent of the number/angle. **Example Use Case:** Find the ratio of the side opposite the given angel to the adjacent side.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

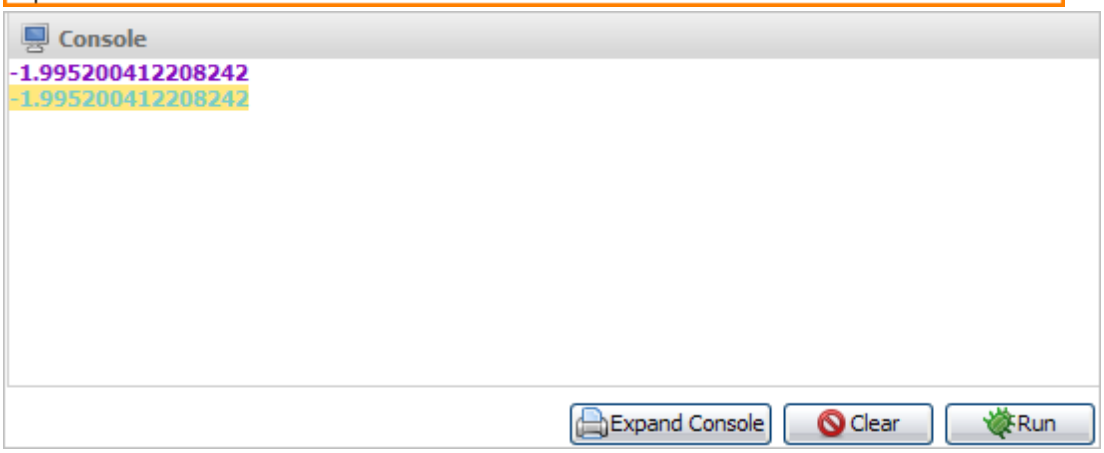
Example of tan(x)

```
sizeofAngle=90.0;

results=tan(sizeofAngle);

print results;

return results;
```



tanh(x)

This function returns the hyperbolic tangent of a number or angle.

#	Parameter	Data Type	Description
1	x	Float	Represents a given float input parameter.

Return Type: Float

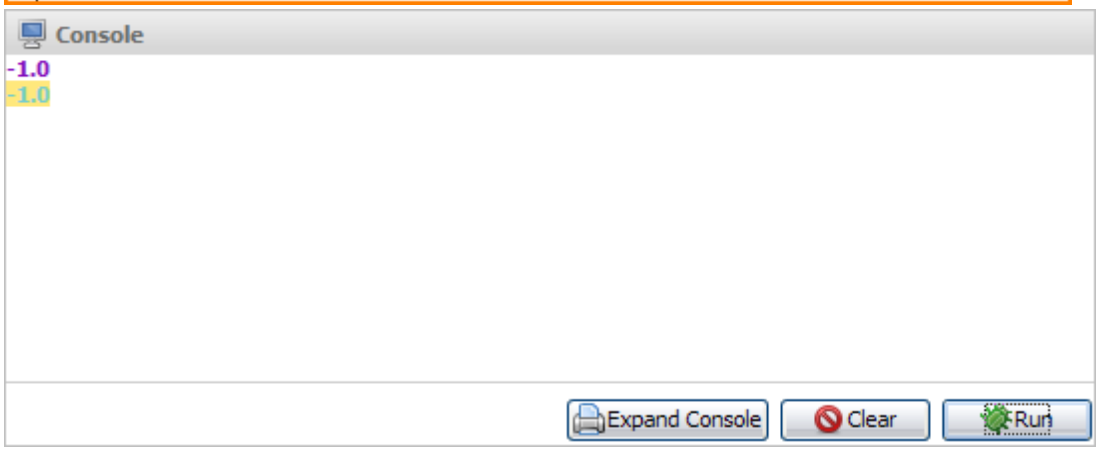
Example of tanh(x)

```
sizeofAngle=-90;

results=tanh(sizeofAngle);

print results;

return results;
```





tanh x = (ex - 1) / (e2x + 1)

RELATED TOPICS

STRING FUNCTIONS

OVERVIEW

String data types allow any type of text character to be returned. However, be aware that not all string functions have return values of the string type.

atof(str) & atoi(str)

The `atof()` and `atoi()` functions convert text that represents a number into either a float or an integer. The syntax is exactly the same. **Example Use Case:** Change a string table value into a float/integer value.

#	Parameter	Data Type	Description
1	str	String	Given input string that is to be parsed into a float or an integer.

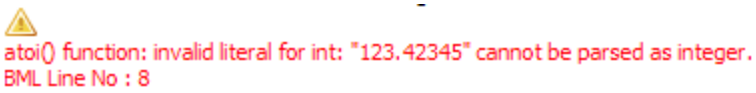
Return Type: Integer

Example of atof(str)/atoi(str) error messages:

- If you attempt to pass an empty string or a string with letters into either `atoi()` or `atof()`, you will see:

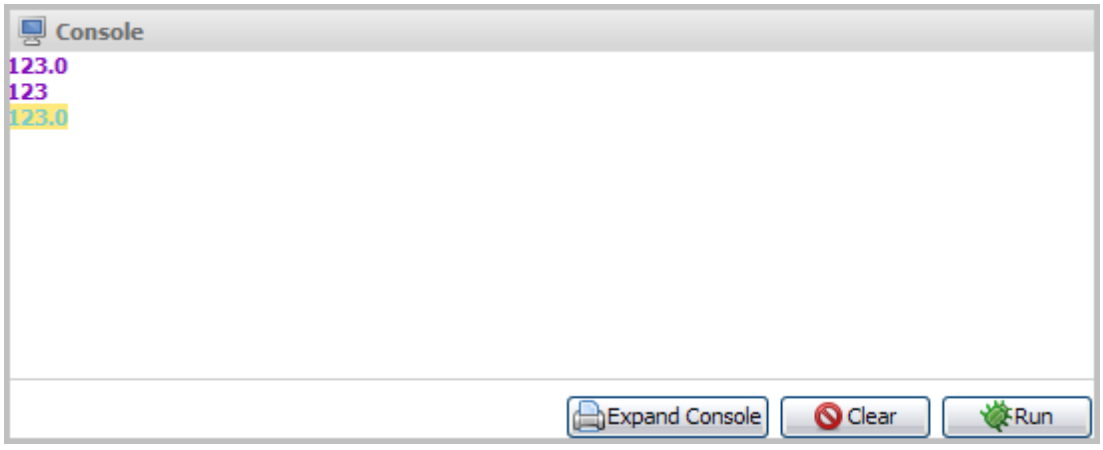


- If you attempt to pass a string with a decimal point ("123.456") into `atoi()`, you will see:



Example of atof(str) and atoi(str)

```
1  strVar_1="123";
2  strVar_2="";
3
4  floatResults=atof(strVar_1);
5          print floatResults;
6
7
8  integerResults=atoi(strVar_1);
9          print integerResults;
10
11 return floatResults;
```





You can't parse a string with a decimal point into an integer.



An empty string will throw an exception error.

---

find(str, substring, [start], [end])

The `find()` function returns the position of a substring within a string.

Example Use Cases:

- Multi-select menu checking
- Find a value or where a value exists within a string
- Powerful when used with a substring function

#	Parameter	Data Type	Description
1	str	String	Represents the given input string.
2	substring	String	Required string parameter that specifies the substring, position of which this function will return.
3	[start]	Integer	Optional: can be used to specify the index at which to begin the search for the substring.
4	[end]	Integer	Optional: can be used to specify the index at which to end the search for the substring.

Return Type: Integer

Example of find(str, substring, [start], [end]) *You will notice that in the example we are performing four different variations of the fin () function:*

- **emptyTest:** We are trying to see what will be returned if we pass two empty strings through the `find()` function.
- **test_2:** what will happen if you try to pass the same two empty strings, but also define the optional start index for the substring.
- **longTest:** shows what is returned when we pass through a string and substring, but begin searching for the substring at an index that is outside of the length of the string.
- **result:** shows what happens if `find()` is looking for a substring and contains both a start and end index.

```
strVar="Pumps for sale.  Pumps are discounted.  Pumps are blue.";
strVar_2="";
subStr="discounted";
subStr_2="";
start_x=5;
start_x2=100;
end_x=100;

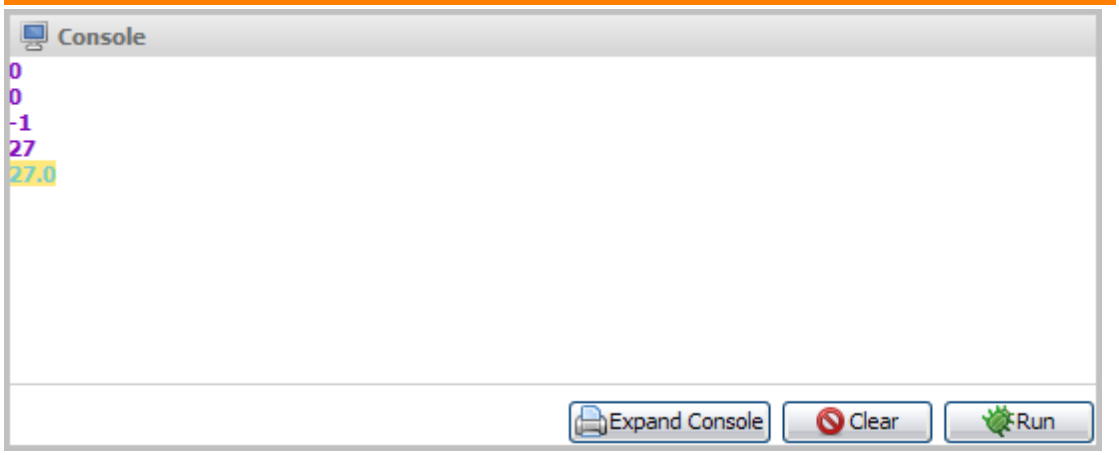
emptyTest=find(strVar_2, subStr_2);
        print emptyTest;

test_2=find(strVar_2, substr_2, start_x);
        print test_2;

longTest=find(strVar, subStr, start_x2);
        print longTest;

results=find(strVar, subStr, start_x, end_x);
        print results;

return results;
```



`Find("", "")` or `find("", "", num>0)` returns 0. `Find("str", "substring", start>length(str))` returns -1.

If not specified, start is 0 and end is length(str).

This function is case-sensitive.

isnumber(str)

`Isnumber(str)` is a Boolean function that returns true when the string is a number and false if it contains other characters. **Example Use Case:** Check if a table cell contains a number. This check needs to be performed to prevent system errors when converting that value to a number.

#	Parameter	Data Type	Description
1	str	String	Represents the given input string .

Return Type: Boolean

Example of isnumber(str)

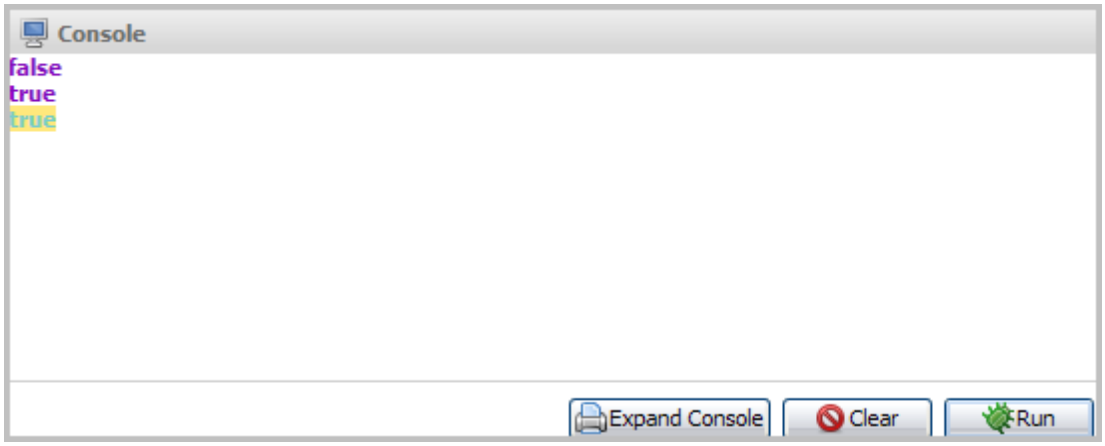
```
strVar_1="12635.51651";
strVar_2="";

emptyResults=isnumber(strVar_2);
        print emptyResults;

results=isnumber(strVar_1);

print results;

return results;
```



If str is an empty string, then the function returns false.

If no string is passed into the function, then a compile time error is thrown.

---

len(str)

The `len()` function will return the length of a string. **Example Use Case:** To constrain the length of an input text field.

#	Parameter	Data Type	Description
1	str	String	Represents the given input string.

Return Type: Integer

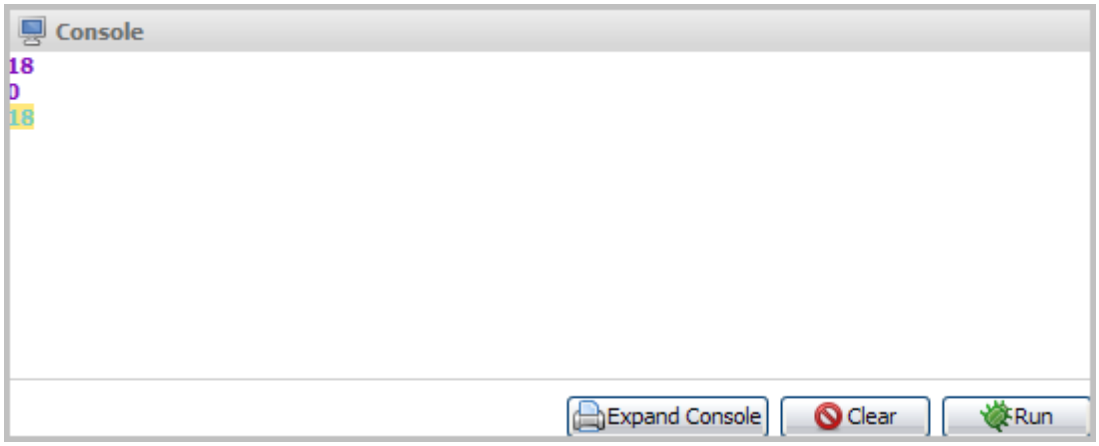
Example of len(str)

```
strVar_1="I like this string";
strVar_2="";

regularResults=len(strVar_1);
        print regularResults;

emptyResults=len(strVar_2);
        print emptyResults;

return regularResults;
```



If str is an empty string, then the function returns a 0.

---

replace(str, old, new, [n])

You will use this function if you want to return a copy of a string, with all occurrences of the old parameter replaced with the new parameter. All occurrences of the old substring are replaced by new. If the optional integer argument is given, only the first n occurrences are replaced.**Example Use Cases:**You must replace multiple subsets of a string.

#	Parameter	Data Type	Description
1	str	String	Represents the given input string.
2	old	String	Specifies the substring that is being replaced.

#	Parameter	Data Type	Description
3	new	String	Specifies the substring that is replacing the old substring.
4	[n]	Integer	Optional: specifies the number of occurrences of the old that must be replaced.

Return Type: Copy of String with replacements made.

Example of replace(str, old, new, [n])

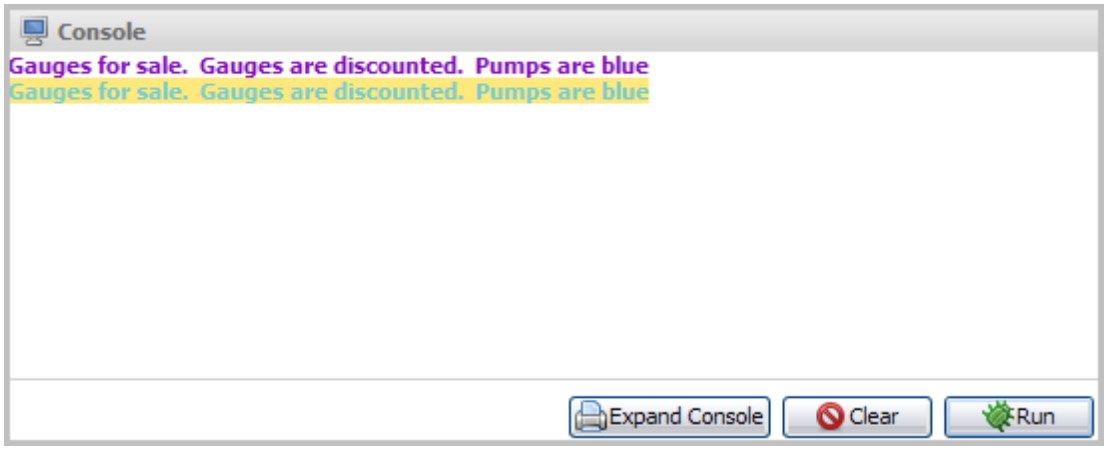
```
strVar="Pumps for sale.  Pumps are discounted.  Pumps are blue";
oldSubStr="Pumps";
newSubStr="Gauges";
intVar=2;

//Replace first two instances of "Pumps" with "Gauges"

results=replace(strVar, oldSubStr, newSubStr, intVar);

print results;

return results;
```



This function is case-sensitive.

Replacing with any empty string throws an exception: ("", "", ""), ("abc", "", "") or (""abc", "", "I").

---

startswith(str, substring) & endswith(str, substring)

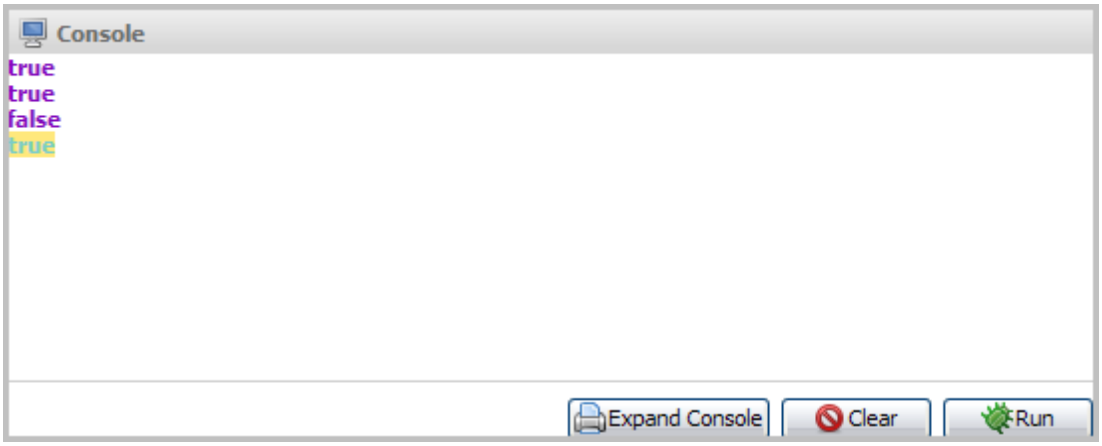
This function, as well as `endswith(str, substring)`, checks whether a string starts or ends with a particular substring. For example, if you wanted to see if the value "abc" starts with "a" or if that same string ends with "c", you would use these functions accordingly.:**Example Use Cases** Determine if a part number starts with a certain text and if a part number contains a specific suffix.

#	Parameter	Data Type	Description
1	str	String	Represents the given input string.
2	substring	String	Endswith() determines if the given input string ends with this substring and startswith() determines if this string begins with this substring.

Return Type: Boolean

Example of startswith(str, substring)/endswith(str, substring)

```
1  strVar_1="I like this string";
2  strVar_2="";
3  subStr="string";
4
5  endsResults=endswith(strVar_1, subStr);
6          print endsResults;
7
8  emptyResults=endswith(strVar_1, strVar_2);
9          print emptyResults;
10
11 startsResults=startswith(strVar_1, subStr);
12         print startsResults;
13
14 return endsResults;
```



We know that the return of `endResults` will be true because the string "I like this string", ends with the substring "string".



An empty string will return as true.



These functions are case-sensitive.

---

string(float(integer(boolean)))

#	Parameter	Data Type	Description
1	Float, Integer, or Boolean	String	Represents the given input that must be converted into a string.

Return Type: String

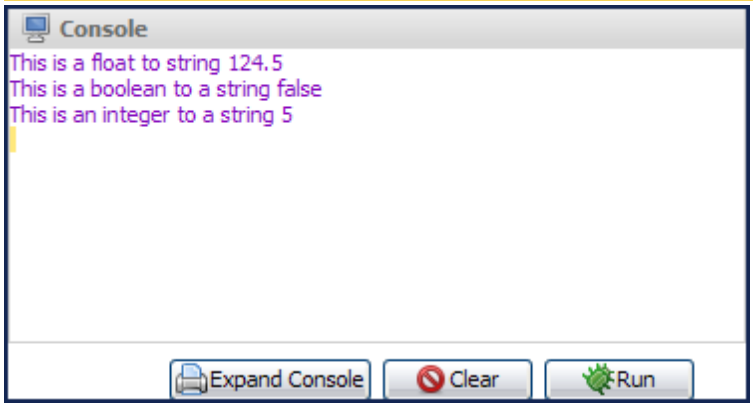
Example of string(float(integer(boolean)))

```
var_1 = 124.5;
var_2 = false;
var_3 = 5;

results = "This is a float to string " + string(var_1);
        print results;

results_2 = "This is a boolean to a string " + string(var_2);
        print results_2;

results_3 = "This is an integer to a string " + string(var_3);
        print results_3;
return "";
```





This function is case-sensitive.



The function will throw a compile time error if a string is passed in as a parameter.

---

substring(str, start, [end])

This function is used to pull a subset of a string.This function is case sensitive. **Example Use Cases:**

- Break a large string down into similar strings
- Test characters of dynamic part numbers
- Determine which characters are at what index

#	Parameter	Data Type	Description
1	str	String	Represents the original string you are going to parse.
2	start	Integer	The index where you will begin parsing the string.
3	[end]	Integer	Optional: Index where you will stop parsing.

Return Type: String

Example of substring(str, start, [end]) *We are performing a number of actions with the substring() function:*

- **subStr:** The first think you should notice is that substring is being populated by using the str, start index and optional [end] index.
- **test_1:** This is using the find function with the substring we just defined.
- **negIndResults:** Here we are giving you an example of what happens when you use a negative start and end index.
- **longStartInd:** This last example shows you what to expect if start is given an index value that is greater than the length of the string.

```
strVar="Pumps and Gauges are discounted.  Pumps are large.";
start_x=10;
end_x=16;
start_1=-29;
start_2=100;
end_1=-19;

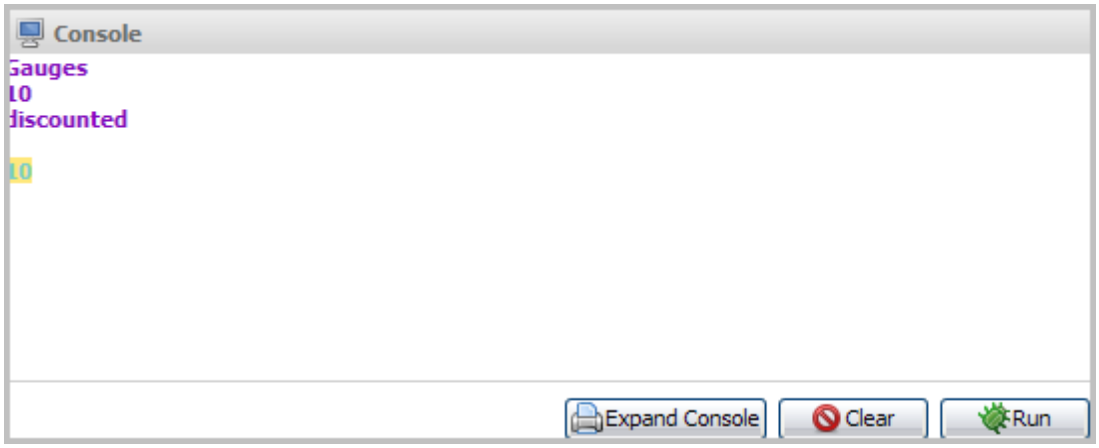
subStr=substring(strVar, start_x, end_x);
        print substr;

test_1=find(strVar, subStr);
        print test_1;

negIndResults=substring(strVar, start_1, end_1);
        print negIndResults;

longStartInd=substring(strVar, start_2);
        print longStartInd;


return test_1;
```





Start and End can have a negative index value.



If start is given an index value that is greater than the length of the string, an empty string will be returned.

---

trim(str)

This function removes the white space from both edges of strings.**Example Use Case** Compare strings with leading or trailing white space.

#	Parameter	Data Type	Description
1	str	String	Represents the given input string.

Return Type: String

Example of trim(str)

```
strVar_1="   12635.51651    ";
strVar_2="";

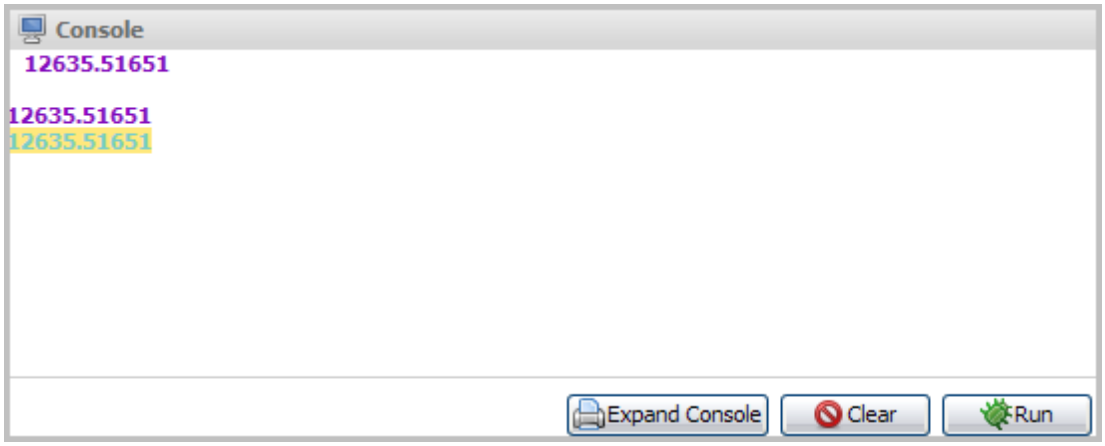
//Print original variable to see the whitespace
print strVar_1;

//Pass an empty string
emptyResults=trim(strVar_2);
        print emptyResults;

//Pass the original variable through the function
results=trim(strVar_1);

print results;

return results;
```



This function is case-sensitive.

If str is empty, then the function returns an empty string.

upper(str) & lower(str)

These functions convert all characters in the text into all lowercase or uppercase letters. **Example Use Cases:**

- Used in table comparisons to make sure you have clean data.
- Ignore case in searches.
- Correcting mistakes made from not following Best Practices during implementation.

#	Parameter	Data Type	Description
1	str	String	Given input string that must be converted to uppercase or lowercase.

Return Type: String

Example of upper(str) & lower(str)

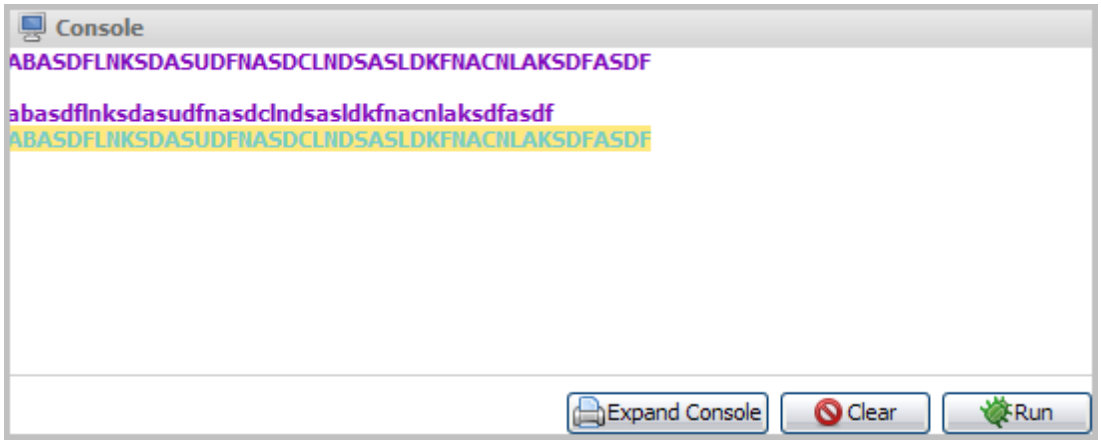
```
strVar_1="ABASDflnksdasudfNASDClndsaSLDkfnaCNlaksdfasdf";
strVar_2="";

upperResults=upper(strVar_1);
        print upperResults;

emptyStringResults=upper(strVar_2);
        print emptyStringResults;

lowerResults=lower(strVar_1);
        print lowerResults;

return upperResults;
```



The code will print three different examples:

- Application of the `upper(str)` function to a typical output string.
- Application of the same function to an empty string.

- Application of the `lower(str)` function to the same string. You will notice a blank space in between the `upper(str)` and `lower(str)` functions. *This represents the return that will occur if you use an empty string.*



This function is case sensitive.



If the string is empty, then the function will return an empty string.

RELATED TOPICS

ADVANCED FUNCTIONS

OVERVIEW

Once you've mastered the standard BML functions, you can move on to the advanced functions. Advanced functions utilize Dictionaries and pull external information from external resources, using data tables and a couple of functions that are specific to either configuration or commerce.

ADMINISTRATION

getarraystr()

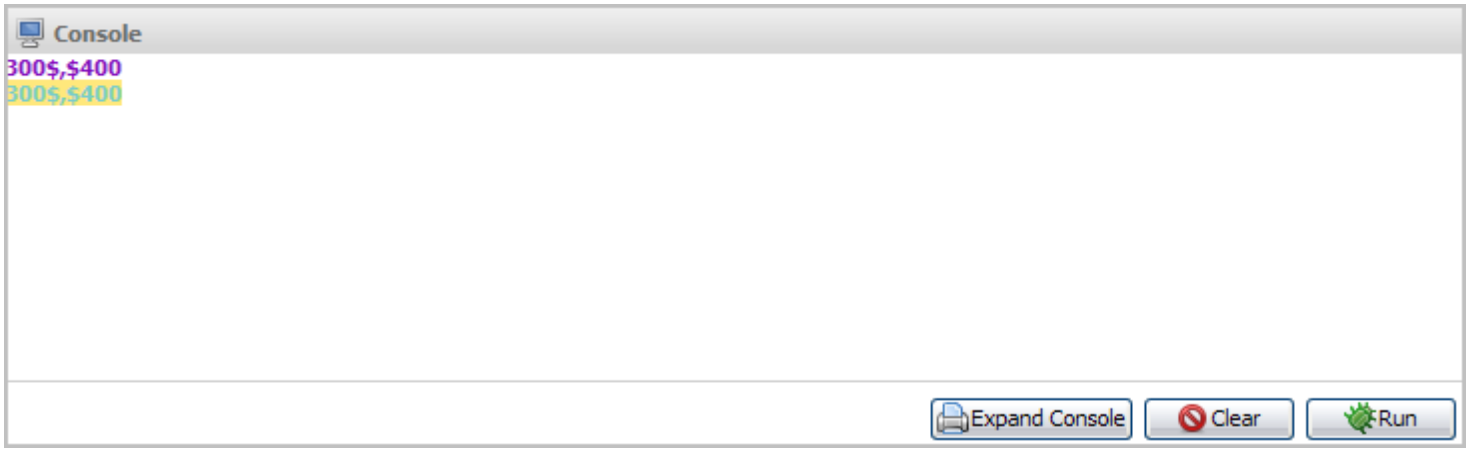
This function returns the delimited string for array attributes with $,$ as the delimiter. This function is only used for configurable attributes.

#	Parameter	Data Type	Description
1	arrayIdentifier	String[]	This is the given input array.

Return Type: String

Example of getararystr()

```
1 pool_length = integer[]{300,400};
2
3 test = getarrayattrstring(pool_length);
4
5 print test;
6 return test;
```



getconfigattrvalue()

This function retrieves the values of configurable attributes in Commerce.

#	Parameter	Data Type	Description
1	[documentNumber]	Integer	Represents the quote number. This is optional.
2	configAttrVarName	String	Variable name of the configurable attribute from which you are retrieving data.

Return Type: String (always returns as a String)



In case of menu attributes, the returned value is the menu item variable name.

Example of getconfigattrvalue()

```
// Sets the line level attribute, lineItemAttribute with the
// value of the configuration attribute, configAttributeValue

retString = "";
for line in line_process{  // loops through all line items

  // make sure that ilne item is a model
  if(line._model_name <> ""){

    // calls function
    configAttributeValue = getconfigattrvalue(line._document_number,
                                "attributVariableName");
    // creates a return string to set line item attribute for all models
    retString = retString + line._document_number + "~lineItemAttribute~"
                + configAttributeValue + "|";
  }
}

return retString;
```



System Attribute _config_attr_info has to be selected as a rule input. If it is not selected and getConfigAttrVal us used, a compile error should be shown to the user.

---

getreasonstatus()

This function returns the status of the reason variable name in an approval sequence.

#	Parameter	Data Type	Description
1	reasonVarName	String	This is the variable name of the reason within the approval sequence.

Return Type: Integer (associated with one of the following statuses):

- **BM_REASON_STATUS_INVALID** : No reason exists in admin reason tree with given variable name.
- **BM_REASON_STATUS_INACTIVE** : Reason exists in admin reason tree but the condition for the reason is not met, therefore no record for it exists in the user-side tree.
- **BM_REASON_STATUS_PENDING** : Reason exists in the user-side reason tree and there are pending approvals.

- **BM_REASON_STATUS_APPROVED** : Reason exists in the user-side tree and it is approved by all approvers, therefore the reason is completely approved.
- **BM_REASON_STATUS_REJECTED** : Reason exits in user-side tree, but has been rejected by at least one approver.

---

isnull()

Evaluates whether a particular Object is null or not. Returns true if argument passed is null.

#	Parameter	Data Type	Description
1	obj	String, Date, Array, Dictionary	This is the variable name of the reason within the approval sequence.

Return: Boolean

Example of isnull()

```
var = "123";

return isnull(var);
```

Return: False

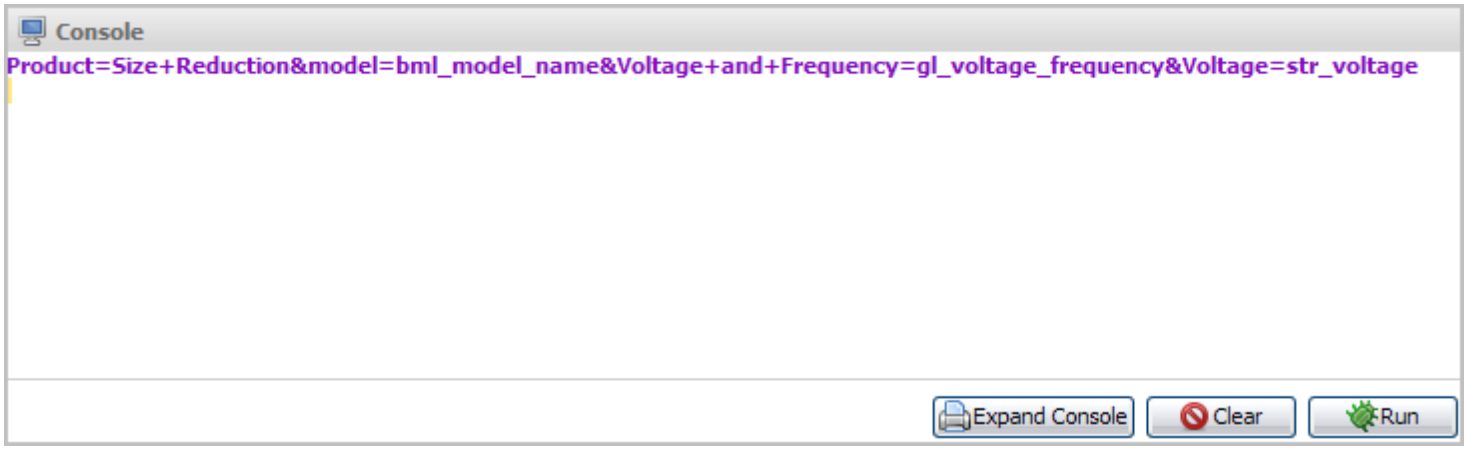
---

makeurlparam({'name ':"value "})

This function is equivalent to creating an ampersand delimited string. URL functions can be used to extract real-time data from an external system. To build a URL that returns relevant data, use a URL variable to define the parameters the URL expects. The `makeurlparam()` function constructs the parameters which are used by other URL access functions.

Example of makeurlparam()

```
params = makeurlparam({'Product':"Size Reduction",
                       'model':"bml_model_name",
                       'Voltage and Frequency':"gl_voltage_frequency",
                       'Voltage':"str_voltage"});
print params;
return "";
```



---

recordset()

The RecordSet data type is a collection of dictionaries. It can also be used as a function in conjunction with BMQL.

Example of RecordSet()

```
rs = recordset();

if(...){
        rs = bmql(query1);
}
else {
        rs = bmql(query2);
}
return rs;
```

There is currently no way to return a `recordSet` result in a Util library function.

---

urldatabyget()

The function retrieves data from a URL address by the HTTP Get Method. BigMachines allows access to Image Server files through File Attachment attribute and `urlDataByGet()` even when the Image Server is secure. You can upload a file to the File Manager and "read" its content in BML by using `urlDataByGet()`. Additionally, an image uploaded to the File Manager can be accessed in the Document Engine through File Manger reference, Hyperlink and External Link. Similarly, file attachments can be accessed in the Document Engine.

Integration Criteria	Value
Size	Varies, but generally small
Frequency	Triggered during workflow
Format	Passes arguments as a URL
Transmission Synchronization	Synchronous

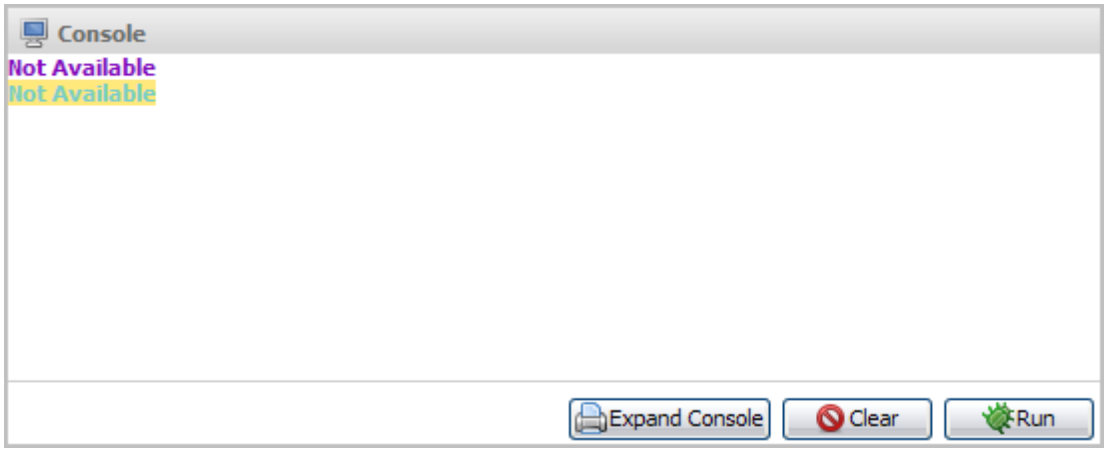
This function is used for RESTful Web Services and is faster than the POST method.

#	Parameter	Data Type	Description
1	URL	N/A	The location of the data you are trying to access.
2	url_param	Dictionary	This represents the makeurlparams() parameters that you created.
3	default_value	String	This is the default value (or error message) displayed if the data cannot be accessed.

Return Type: String

Example of urldatabypost(URL, url_param, default_value)

```
1  //This provides inputs to Perl program in http://resources.bigmachines.com/cgi-bin/myCompanylead
2  //Program output is displayed to user. If data can't be accessed, then an error message will app
3
4  params = makeurlparam({'product':"Size Reduction",
5          'model':"_bm_model_name",
6          'Voltage and Frequency':"gl_voltage_frequency",
7          'Voltage':"sr_voltage"});
8
9  data = urldatabyget("http://resources.bigmachines.com/cgi-bin/myCompanyleadtime.pl", params, "No
10
11 print data;
12 return data;
```



urldatabypost()

This function retrieves data from a URL address by the HTTP Post Method. This function passes parameters separately from the URL. It is useful when handling sensitive data and/or larger requests. It allows for encrypted requests. This is used for SOAP Web Services. For more information, see Using SOAP with BML.

Integration Criteria	Value
Size	Varies, but generally small
Frequency	Triggered during workflow
Format	Passes arguments as XML (generally)
Transmission Synchronization	Synchronous

Best Practice is to use the File Manager to hold the form POST.

#	Parameter	Data Type	Description
1	URL	N/A	The location of the data you are trying to access.
2	url_param	Dictionary	This represents the makeurlparam() parameters you created.
3	default_value	String	This is the default value (or error message) displayed if the data cannot be accessed.

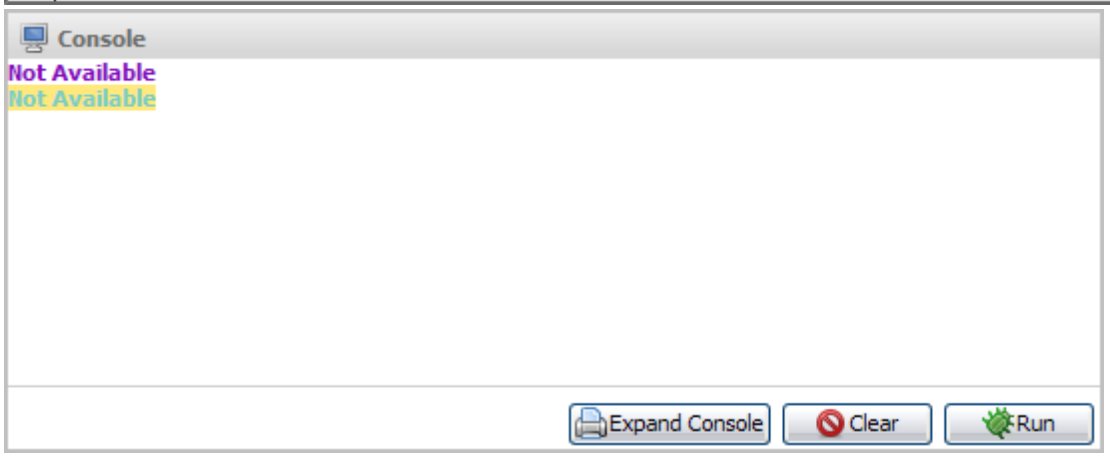
Return Type: String

BigMachines allows additional parameters that can be passed to `urldatabypost` to pass header values. **Example:**

```
headers = dict("string");
put(headers, "SoapAction Key", "SoapAction Value");
return urldatabypost("https://cpluto.bigmachines.com/httpreceiver", "", "", headers);
```

Example of urldatabypost(URL, url_param, default_value)

```
1  //This provides inputs to Perl program in http://resources.bigmachines.com/cgi-bin/myCompanylead
2  //Program output is displayed to user. If data can't be accessed, then an error message will app
3
4  params = makeurlparam({'product':"Size Reduction",
5          'model':"_bm_model_name",
6          'Voltage and Frequency':"gl_voltage_frequency",
7          'Voltage':"sr_voltage"});
8
9  data = urldatabyget("http://resources.bigmachines.com/cgi-bin/myCompanyleadtime.pl", params, "No
10
11 print data;
12 return data;
```



We see, just as the function suggests, this information is not available.

Sample Request:

```
headers = dict("string");
put(headers, "Content-Type", "text/xml");

requestURL="https://"+_system_supplier_company_name+".bigmachines.com/v1_0/receiver";

soapHeader =
      "<soapenv:Header>" +
         "<bm:userInfo xmlns:bm='urn:soap.bigmachines.com'>" +
            "<bm:sessionId>" + _system_user_session_id + "</bm:sessionId>" +
         "</bm:userInfo>" +
         "<bm:category xmlns:bm='urn:soap.bigmachines.com'>Data Tables</bm:category>" +
         "<bm:xsdInfo xmlns:bm='urn:soap.bigmachines.com'>" +
            "<bm:schemaLocation>https://"+_system_supplier_company_name+
            ".bigmachines.com/bmfsweb/sprint/schema/v1_0/datatables/Menu_Pricing.xsd</bm:schemaLocation>" +
         "</bm:xsdInfo>" +
      "</soapenv:Header>";

//Deploy the data table
soapBody =
      "<soapenv:Body>" +
      "<bm:deploy xmlns:bm='urn:soap.bigmachines.com'>" +
      "<bm:DataTables bm:table_name='Menu_Pricing'/>" +
      "</bm:deploy>" +
      "</soapenv:Body>";

soap = "<?xml version='1.0' encoding='UTF-8'?><soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envel
      soapHeader + soapBody + "</soapenv:Envelope>";

print soap;

result = urldatabypost(requestURL, soap, "NA", headers);
```

RELATED TOPICS

DICTIONARIES

OVERVIEW

Dictionaries are used to store key value pairs. Their purpose is similar to that of an array or data table. A dictionary key is used to loop up a value. Dictionaries allow users to store and retrieve data quickly and easily. Once you have created a dictionary, values are stored using a key. You can attach a value with any data type to a dictionary, except another dictionary. When you declare a dictionary, ensure to declare what data type is needed.

ADMINISTRATION

dict(dictType)

This function is used to create a dictionary of a specific data type.

#	Parameter	Data Type	Description
1	**dictType**	string, integer, float, string[], integer[], float[], string[][], integer[][], float[][], boolean or date	Specify the data type of the dictionary.

Return Type: Dictionary

---

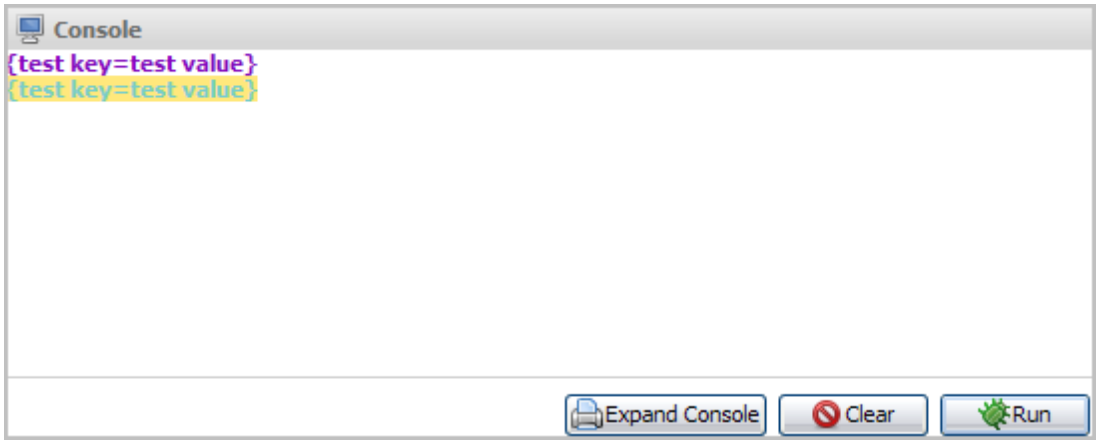
put(dictIdentifier, key, value)

Once your dictionary has been created, you can define keys and values to put into the dictionary.

#	Parameter	Data Type	Description
1	**dictIdentifier**	Dictionary	Identifies the dictionary you are adding values to.
2	**key**	String, Integer, or Float	Unique identifier for the value parameter.
3	**value**	String, Integer, or Float	Data value that relates to the unique key identifier.

put(dictIdentifier, key, value) Example:

```
1 myDictionary = dict("string");
2
3 put (myDictionary, "test key", "test value");
4
5 print myDictionary;
6 return myDictionary;
```



---

get(dictIdentifier, key)

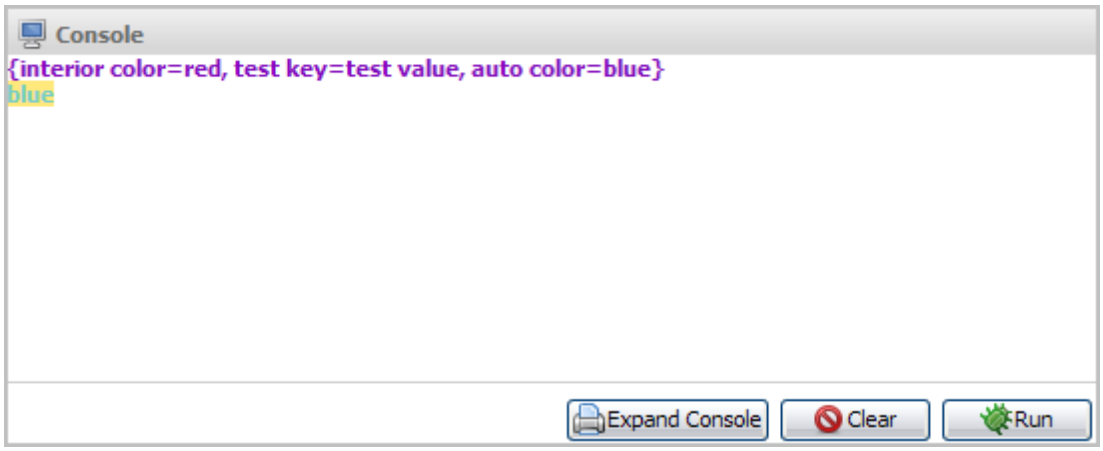
This function will retrieve the value of the provided key from the dictionary.

#	Parameter	Data Type	Description
1	**dictIdentifier**	Dictionary	Identifies the dictionary you are retrieving values from.
2	**key**	String, Integer, or Float	Unique identifier for the value parameter.

Return Type: The return type is based on the dictionary data type.

get(dictIdentifier, key) Example: *Below, three separate keys and values have been defined and placed within the same dictionary.*

```
1  myDictionary = dict("string");
2
3  put(myDictionary, "test key", "test value");
4  put(myDictionary, "auto color", "blue");
5  put(myDictionary, "interior color", "red");
6
7  result = get(myDictionary, "auto color");
8
9  print myDictionary;
10 return result;
```



containskey(dictIdentifier, key)

This is a dictionary function that checks to see if the key is found in the dictionary.

#	Parameter	Data Type	Description
1	**dictIdentifier**	Dictionary	Identifies the dictionary being searched to find if it contains the key you have identified.
2	**key**	String, Integer, or Float	Unique identifier for the value parameter.

Return Type: Boolean

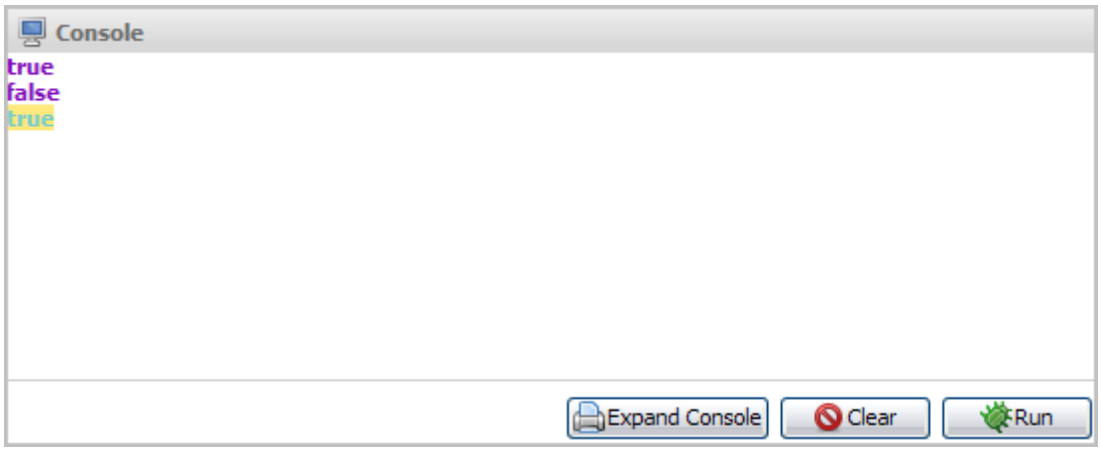
containskey(dictIdentifier, key) Example:

```
myDictionary = dict("string");

put(myDictionary, "test key", "test value");
put(myDictionary, "auto color", "blue");
put(myDictionary, "interior color", "red");

result = containskey(myDictionary, "auto color");
        print result;
test_1 = containskey(myDictionary, "purple");
        print test_1;

return result;
```



keys()

This allows administrators to retrieve an unordered String Array of all keys found within a Dictionary. If the Dictionary does not contain any keys, an empty array is returned.

Refer to the Function Wizard for syntax examples.

NOTES



Dictionaries have better speed and performance than performing table cells.

RELATED TOPICS

FUNCTION EDITOR BASICS

OVERVIEW

While there are notable differences between the function editors found in the BigMachines application, there are some common characteristics as well.

Button Definitions



You will run across a few buttons located at the bottom of the editor pane:

- The **Translations**button translates BigMachines content and product templates into the following languages: French, German, and Spanish.
- The **Validate** or **Check** button checks your code for errors.
- The **Apply** button saves your code and closes the function editor.
- The **Update** button allows you to save your code and continue editing.
- The **Back** button returns you to the previous page.

Syntax Colors

You will notice that the code below is displayed in different colors:

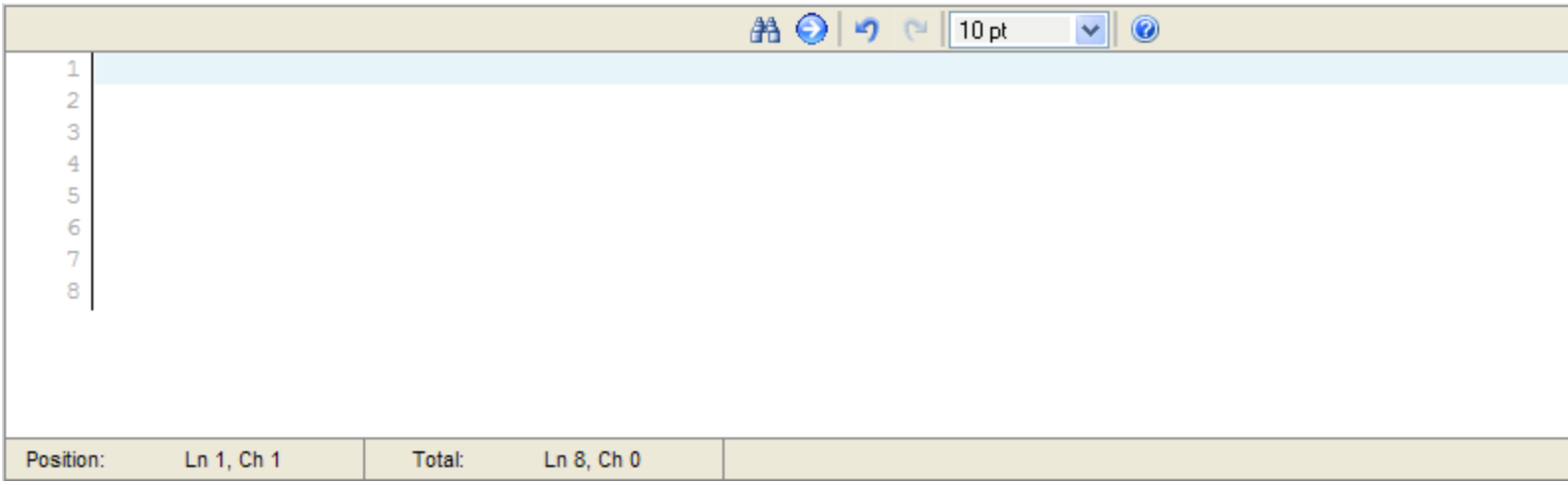
```
1  //Sets line level attribute, lineItemAttribute, w/ value of configuration attribute, configAttr
2
3  retString = "";
4  for line in line_process{ //Loops through all the line items.
5  //Ensure that line item is a model.
6          if(line._model_name <> ""){
7          //Calls function
8          configAttributeValue = getconfigattrvalue (line._document_number, "attributeVariableNam
9          //Creates a return string to set line item attribute for all models.
10         retString = retString + line._document_number + "~lineItemAttribute~"
11                 + configAttributeValue = "|";
12         }
13 }
14 return retString;
```

COLOR	DEFINITION
Black	Variable names, float and integer variables
Pink	Numeric Operators
Aqua	Literals or Data Types (string, float, integer, boolean, etc)

COLOR	DEFINITION
Blue	Functions and string variables
Green	Conditional Functions
Gray	Comments

Script Definition Area

BML can be typed directly into the Script Definition Area. You will want to save your changes after writing/editing the script. **Example:**



Notice that the bar across the bottom represents the character position in the code and the total number of characters.



The box on the left defines the position of the character based on its line and index. The box to the right gives you the actual number of characters in each line.



Clicking on the **Binoculars** will call a "search and replace" pop-up dialog if you need to find and/or replace part of the script.





Use the toolbar above to undo/redo changes, change the font or jump to a specific line in the script.

You can also click on the blue arrow to jump to a specific line in the script.

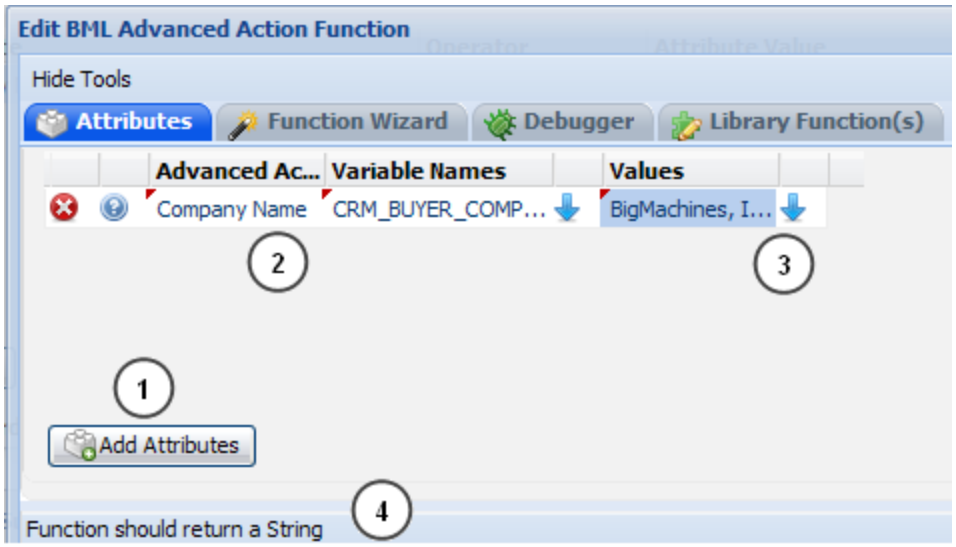
Character indexes begin with 0.

Attributes & Actions

In the function editor, you can select attributes to include within your code. These attributes are specific to the product family that you are working in. For example, when you are in Configuration, you can select from a standard product line, model, account, user or configurable attributes.

Adding an Attribute

1. Select **Add Attributes** in the **Attributes** tab. *This adds an editable row to the Attributes list.*
2. Navigate through the drop-down attributes menu to find the desired attribute **OR**start typing *(the attribute list will filter for you)*.



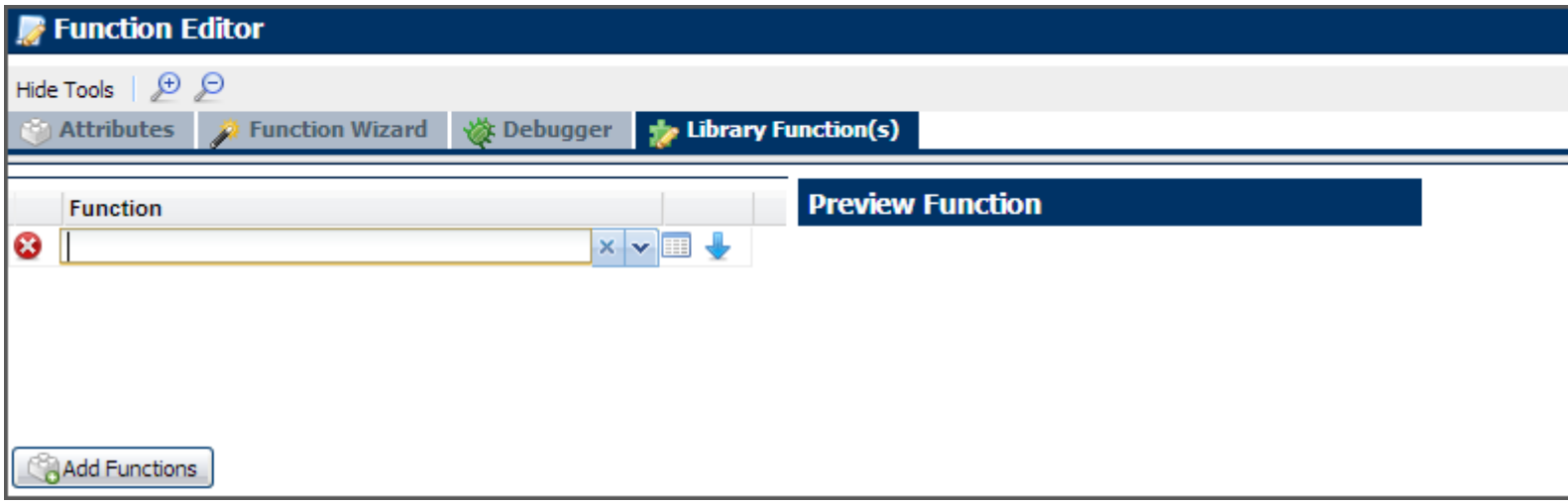
   - The proper **Variable Name** will automatically populate.
3. Click the drop-down list in the first column and navigate through the attributes list to find the desired attribute**OR**click and start typing *(the attribute list will filter for you)*.
   - The proper **Variable Name** will automatically populate.
4. Navigate through the drop-down menu and choose the desired value for the selected attribute.

5. Select the blue arrow to insert the attribute value into the Script Definition Area.

> The return type of your code will be displayed in between the section with the header tabs and the script definition area.
>
> The action attribute values you select will determine the return type of your code.

Library Function(s): Function to Function Calls

Util and Commerce Library Function Editors utilize Function to Function calls. Function to Function calls allow administrators to compartmentalize BML when dealing with complicated configuration or quoting scenarios. This feature will assist with the organization of BML and provides a solution to the compiled Java class size-limit issue. *Function to function calls mimic the behavior of a Modify function calling a Library function.*



Adding a Function to Function Call

1. Open the **Function** drop-down menu.
2. Select a Library Function to **Insert into BML**. The options are:
   - **Commerce Library functions:** Only available in Commerce Library Function Editors. Commerce Library functions may only be called by Commerce Library functions.
   - **Util Library functions:** Available in both Commerce and Util Library Function Editors.
3. Once the function is viewed using the **Preview Function**, click **Insert into BML**blue arrow button.

4. When finished, click one of the following buttons:



The Function to Function link appears on the Related Rules page. *When it is referenced by other Utils it will be displayed on the Related Rules page.*

Recursive validation is performed during the following for Util and Commerce Library Functions: Validate, adding, applying, or updating a function.

Util and Commerce Library functions cannot self-reference. *Recursive calling of the same Util and Commerce Library functions will fail and result in a compilation error when called at any point in the reference chain. Util and Commerce Library functions will not appear in the Import list for themselves.*

RELATED TOPICS

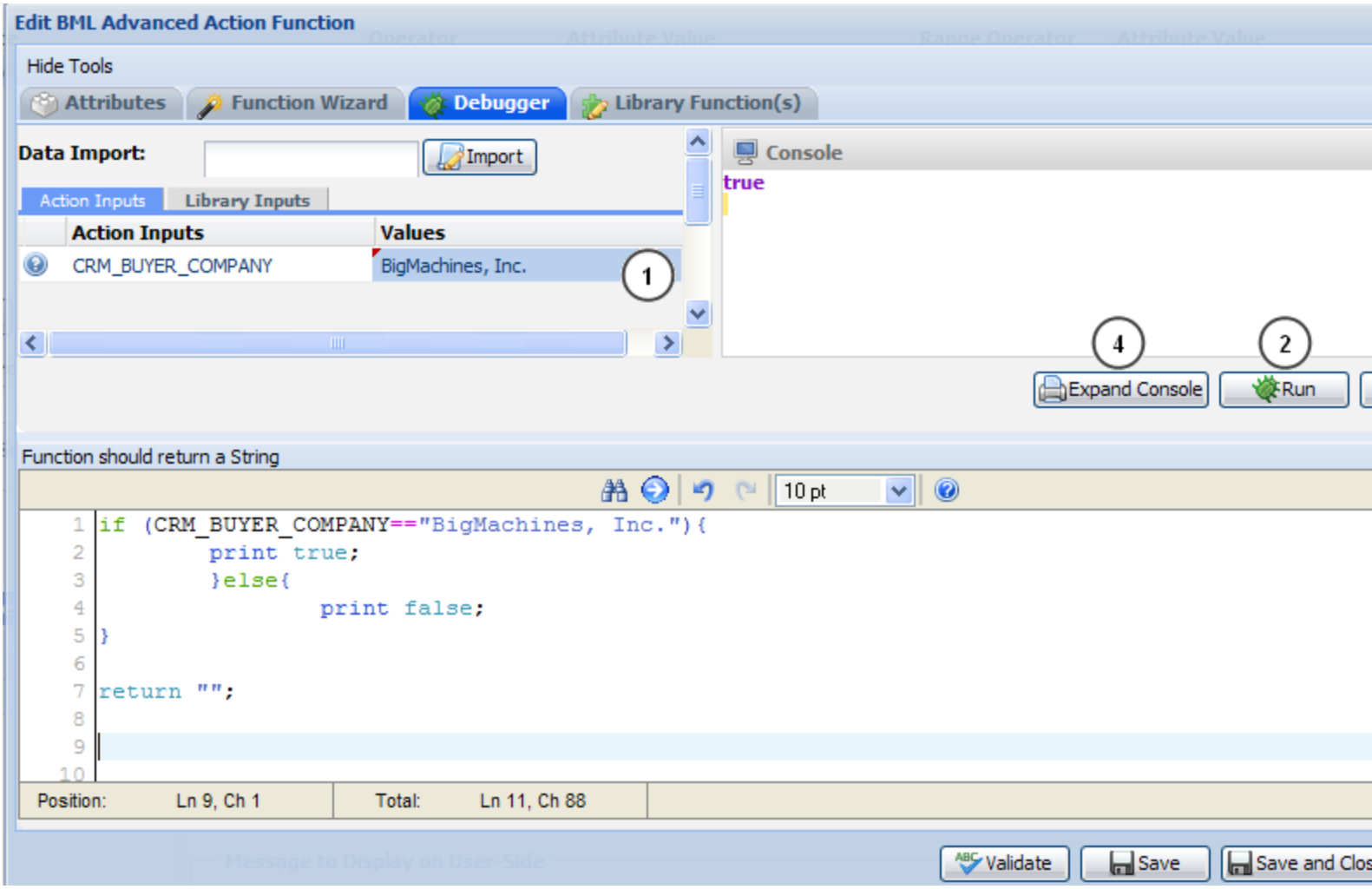
DEBUGGER VARIATIONS

OVERVIEW

The Debugger tab allows the users to debug the advanced function. You can test the code by providing different values of the attributes/parameters as inputs in the debugger. The Debugger varies quite a bit depending on where you are in the application.

Configuration Debugger

Begin in the **Input Section**, located in the upper left-hand corner for the **Function Editor**. *You should have already added your attributes/parameters under the **Attributes** tab. Those will appear in the input section under **Action Inputs**.*

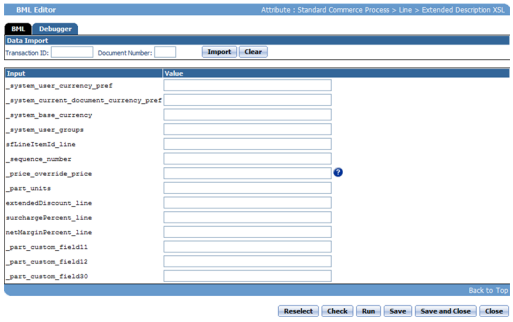


1. Select the corresponding cell in the **Value** column and choose a value for the attribute using the drop-down list **OR** by typing a value into a cell.
2. Select **Run** to run the function. *The result can be seen in the **Console** section.*
3. Select **Clear** to remove the **Value(s)** for your attribute(s) and clear the **Console**.
4. Select **Expand Console** to hide the **Input Section** of the **Debugger**.

If there is an error in the syntax, an error message will appear ABOVE the tabs.

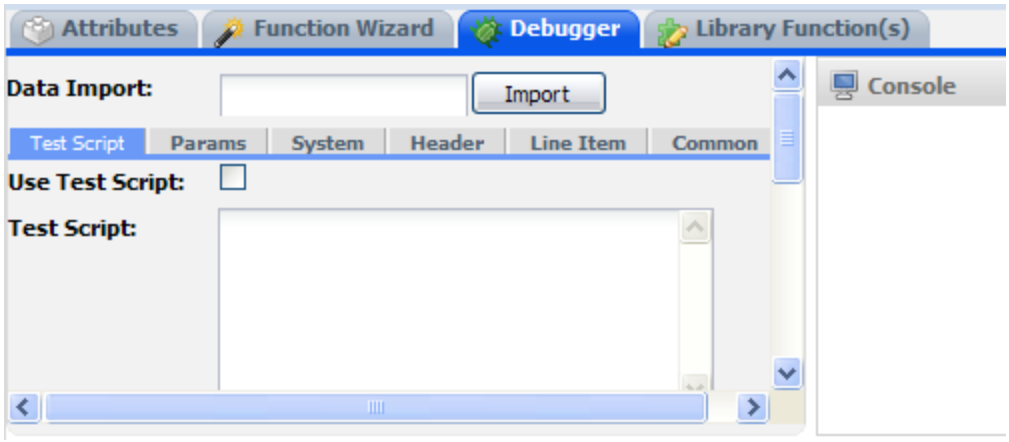
Commerce Debugger

This will eventually be phased out, but there are still many applications and versions out there that still use this. This Debugger requires users to select at least one variable and then use a **Transaction ID** number to import the **Values** for each attribute you selected. You could also manually enter these values.



Commerce Library

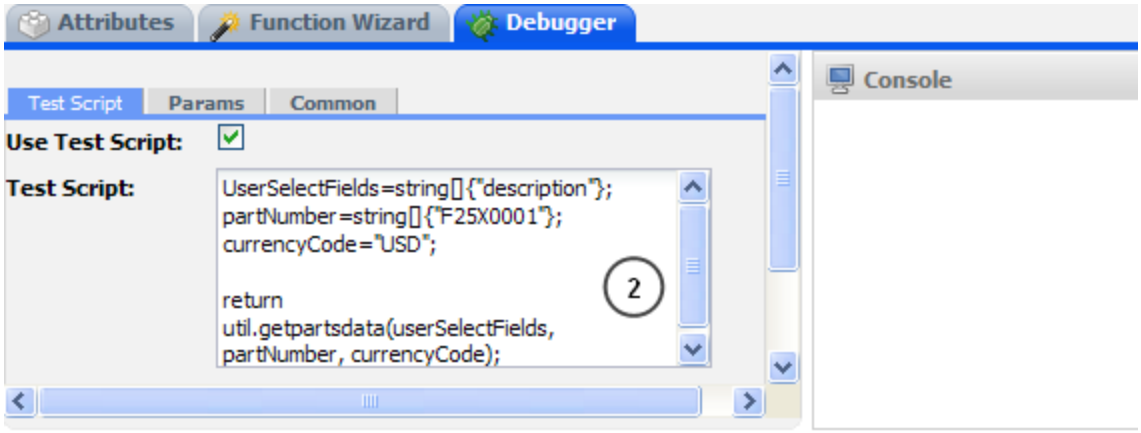
There are a number of sub-headings that can be used under the **Commerce Library Debugger** tab. For more information, see Add a Commerce Library F(x).

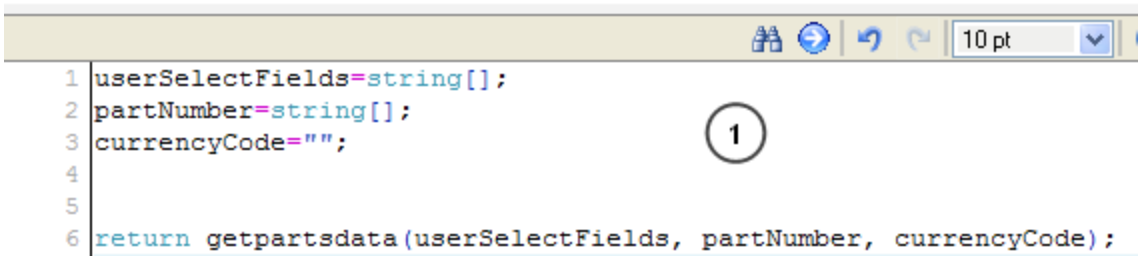


Test Script: Provides a way to test a BML Library function when array type attributes are input parameters. Within the Test Script, you can populate your own complex data objects and pass it to the library functions to analyze the results. *Test Scripts can also be used to compute multiple iterations of the library functions and print each.*

Example of Test Script:

1. Create values for the parameters used in this function.
2. View the return statement, it is written as follows: `return.util.function_name(parameters).`





Term	Definition
Param	If you defined parameters in the header section, they will appear here for Debugging.
System	This section lists the attributes added under the System Attributes section. They will appear here so the parameter values can be populated/changed when debugging. You can populate values by double-clicking the Value section next to the desired attribute(s).
Header	This section lists the attributes added under the Main Document Attributes section. You can add values by double-clicking the Value section next to the attribute(s) of your choice.
Line Item	This section lists the attributes added under the Sub-Document Attributes section. You can populate values for debugging by double-clicking the Value section next to the desired attribute(s).
Common	In the Common Attribute section, click Add to access the common attributes that are available across the system. Example: suppose the function requires you to check the user type of the logged-in user, choose the user type, Common Attribute, and reference it in your function.

Finding Transaction ID

In order to use the Debugger in Commerce, you must reference a quote.

1. Go to the **Quote Manager**.
2. Select a pending quote and its quote number.
3. Select **View > Source** under your address bar.
4. Type **CTRL + F** to open a **Find** box.
5. Search for "id" to locate the **Transaction ID**.



Depending on your implementation, you may be able to find it under a tab titled **Integration**. *Click on the **Integration** tab and the **Transaction ID** will be in a text box, located at the right of the screen.*

RELATED TOPICS

FUNCTION EDITOR BASICS

OVERVIEW

While there are notable differences between the function editors found in the BigMachines application, there are some common characteristics as well.

Button Definitions



You will run across a few buttons located at the bottom of the editor pane:

- The **Translations**button translates BigMachines content and product templates into the following languages: French, German, and Spanish.
- The **Validate** or **Check** button checks your code for errors.
- The **Apply** button saves your code and closes the function editor.
- The **Update** button allows you to save your code and continue editing.
- The **Back** button returns you to the previous page.

Syntax Colors

You will notice that the code below is displayed in different colors:

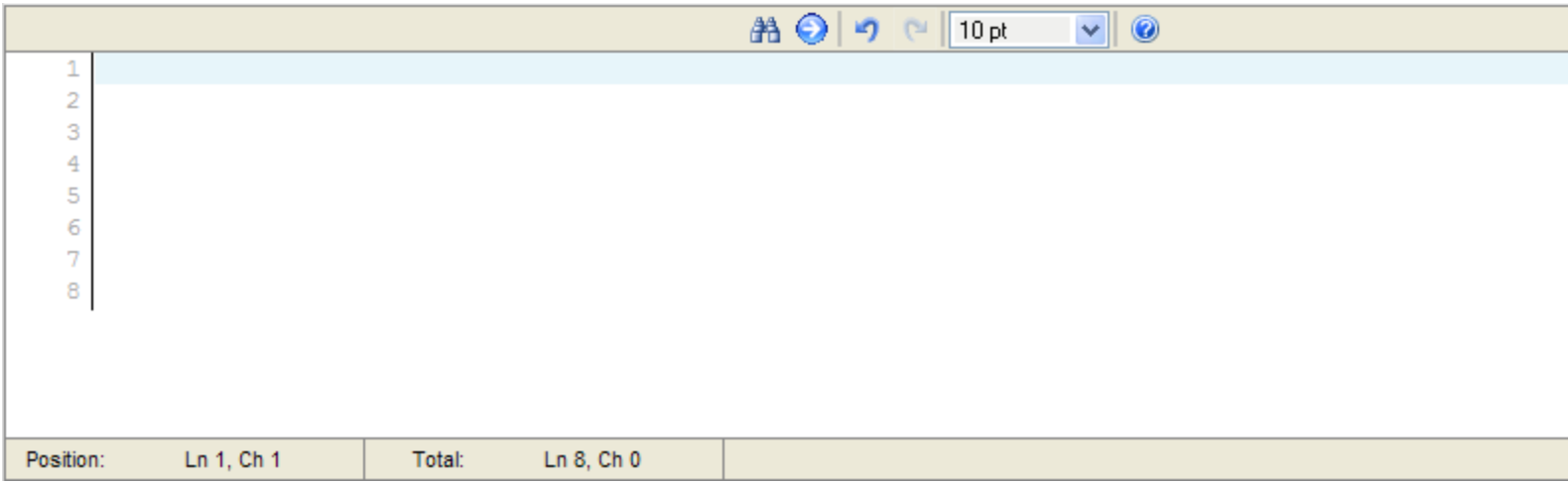
```
1  //Sets line level attribute, lineItemAttribute, w/ value of configuration attribute, configAttr
2
3  retString = "";
4  for line in line_process{ //Loops through all the line items.
5  //Ensure that line item is a model.
6          if(line._model_name <> ""){
7          //Calls function
8          configAttributeValue = getconfigattrvalue (line._document_number, "attributeVariableNam
9          //Creates a return string to set line item attribute for all models.
10         retString = retString + line._document_number + "~lineItemAttribute~"
11                 + configAttributeValue = "|";
12         }
13 }
14 return retString;
```

COLOR	DEFINITION
Black	Variable names, float and integer variables
Pink	Numeric Operators
Aqua	Literals or Data Types (string, float, integer, boolean, etc)

COLOR	DEFINITION
Blue	Functions and string variables
Green	Conditional Functions
Gray	Comments

Script Definition Area

BML can be typed directly into the Script Definition Area. You will want to save your changes after writing/editing the script. **Example:**



Notice that the bar across the bottom represents the character position in the code and the total number of characters.



The box on the left defines the position of the character based on its line and index. The box to the right gives you the actual number of characters in each line.



Clicking on the **Binoculars** will call a "search and replace" pop-up dialog if you need to find and/or replace part of the script.





Use the toolbar above to undo/redo changes, change the font or jump to a specific line in the script.

You can also click on the blue arrow to jump to a specific line in the script.

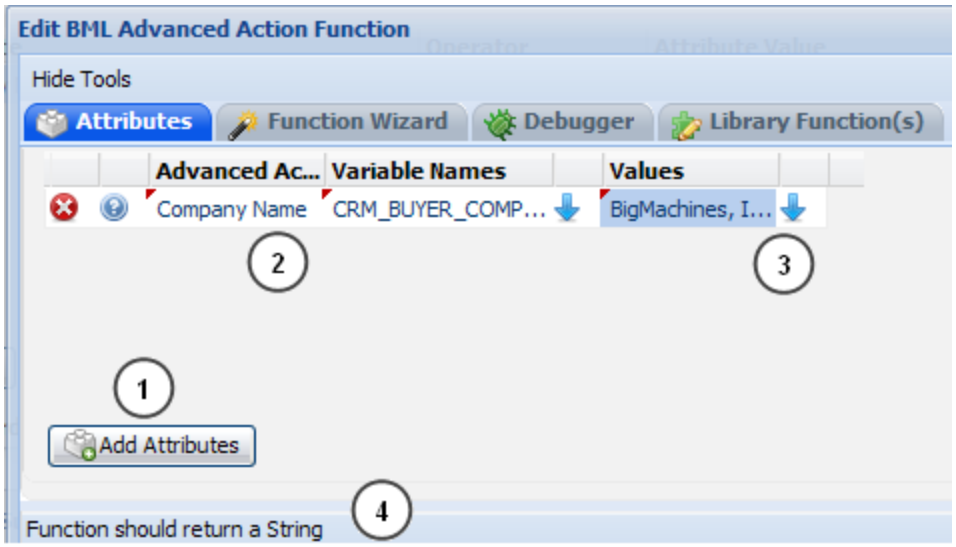
Character indexes begin with 0.

Attributes & Actions

In the function editor, you can select attributes to include within your code. These attributes are specific to the product family that you are working in. For example, when you are in Configuration, you can select from a standard product line, model, account, user or configurable attributes.

Adding an Attribute

1. Select **Add Attributes** in the **Attributes** tab. *This adds an editable row to the Attributes list.*
2. Navigate through the drop-down attributes menu to find the desired attribute **OR**start typing *(the attribute list will filter for you)*.



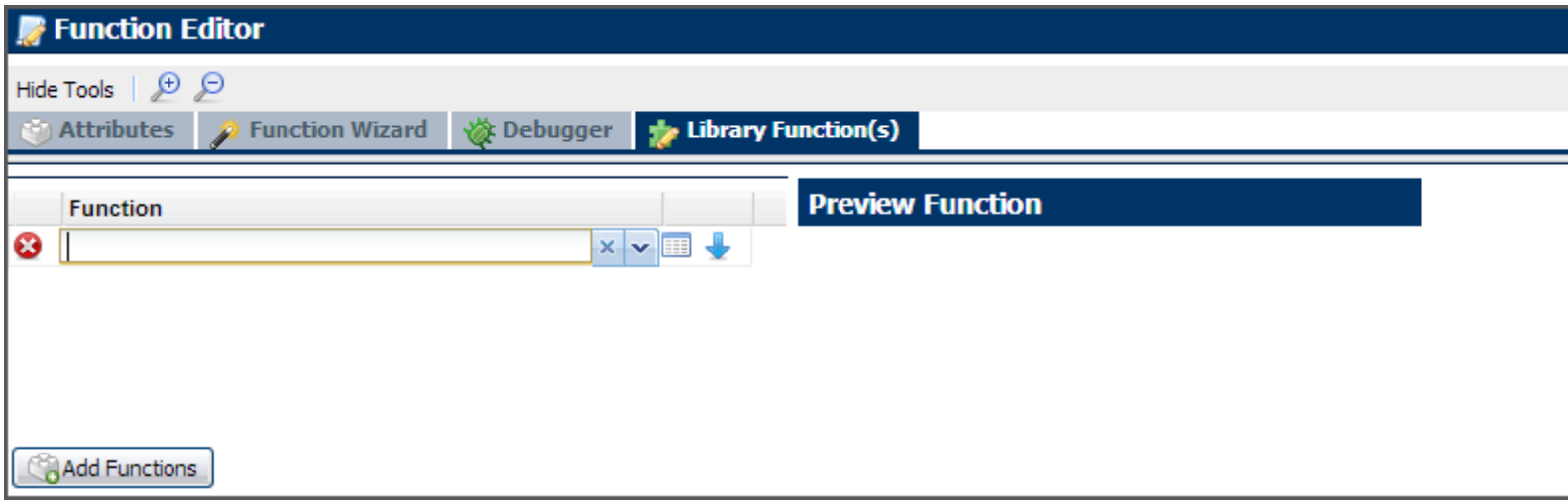
   - The proper **Variable Name** will automatically populate.
3. Click the drop-down list in the first column and navigate through the attributes list to find the desired attribute**OR**click and start typing *(the attribute list will filter for you)*.
   - The proper **Variable Name** will automatically populate.
4. Navigate through the drop-down menu and choose the desired value for the selected attribute.

5. Select the blue arrow to insert the attribute value into the Script Definition Area.

> The return type of your code will be displayed in between the section with the header tabs and the script definition area.
>
> The action attribute values you select will determine the return type of your code.

Library Function(s): Function to Function Calls

Util and Commerce Library Function Editors utilize Function to Function calls. Function to Function calls allow administrators to compartmentalize BML when dealing with complicated configuration or quoting scenarios. This feature will assist with the organization of BML and provides a solution to the compiled Java class size-limit issue. *Function to function calls mimic the behavior of a Modify function calling a Library function.*



Adding a Function to Function Call

1. Open the **Function** drop-down menu.
2. Select a Library Function to **Insert into BML**. The options are:
   - **Commerce Library functions:** Only available in Commerce Library Function Editors. Commerce Library functions may only be called by Commerce Library functions.
   - **Util Library functions:** Available in both Commerce and Util Library Function Editors.
3. Once the function is viewed using the **Preview Function**, click **Insert into BML**blue arrow button.

4. When finished, click one of the following buttons:



The Function to Function link appears on the Related Rules page. *When it is referenced by other Utils it will be displayed on the Related Rules page.*

Recursive validation is performed during the following for Util and Commerce Library Functions: Validate, adding, applying, or updating a function.

Util and Commerce Library functions cannot self-reference. *Recursive calling of the same Util and Commerce Library functions will fail and result in a compilation error when called at any point in the reference chain. Util and Commerce Library functions will not appear in the Import list for themselves.*

RELATED TOPICS

FUNCTION WIZARD

OVERVIEW

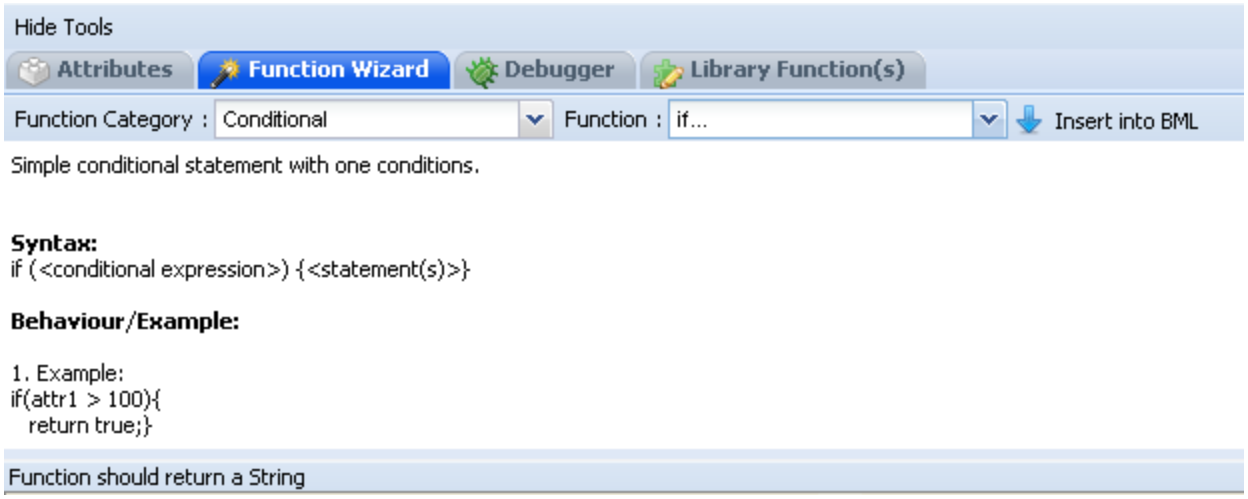
The Function Wizard contains a list of pre-defined functions that are available for use in advanced scripting. There are some variations in the Function Editor, depending on where you are located within the application: Commerce, Configuration, Commerce Library, or Util Library.

Configuration/Util Library Function Editor

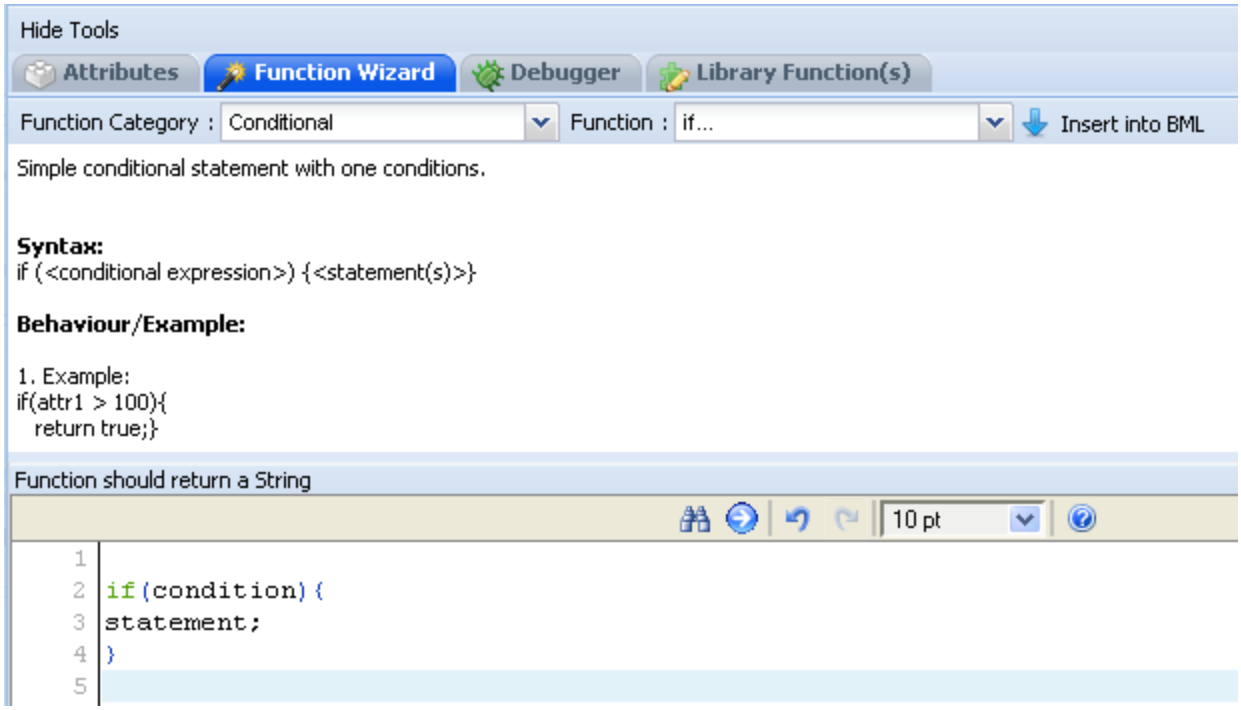
The only difference between the Configuration Function Editor and that in the Util Library is that the latter doesn't have a tab labeled Library Functions. This is because your library functions are created in the Util Library.

Using the Function Wizard in Configuration

1. Open the **Function Wizard** tab.
2. Choose a **Function Category** and a **Function**name. *You will see syntax help appear just below the toolbar.*



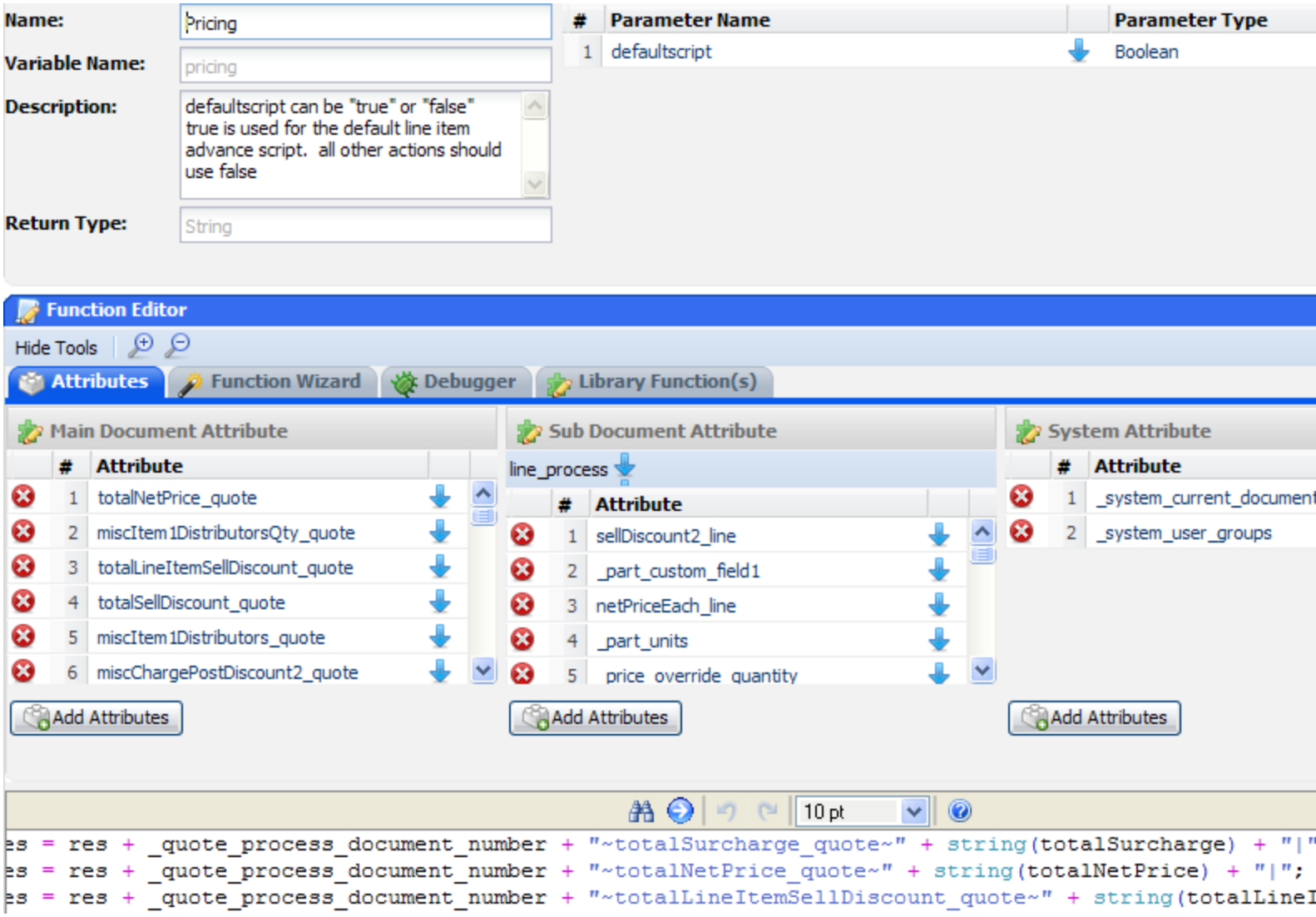
3. Select **Insert into BML** to insert the function into your Script Definition Area.



4. Review the function populated in the script definition area.

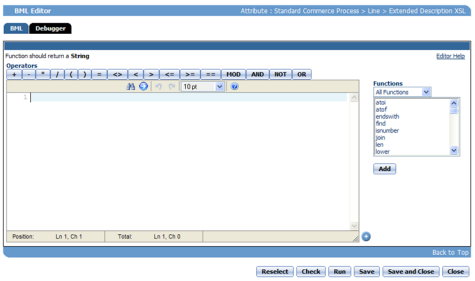
Commerce Library Function Editor

The commerce library function editor looks similar to the one in Configuration, however, you will notice that the Attributes section is quite different. It allows you to select attributes for the main document, sub-document, and system attributes.



Commerce Function Editor

You may come across a BML editor in Commerce that looks like this:





Using this editor, you will have selected your attributes on a previous screen. Your operators are all located on a toolbar across the top. You can choose your functions from a drop-down menu.

RELATED TOPICS

ACCOUNTS

OVERVIEW

Accounts is a tool for supporting collaborative customer management in the BigMachines system. Customer records can be individually created in the Accounts database or they can be bulk uploaded into the system. Once customer records exist in Accounts, they are accessible to sales agents, channel agents and full access users.

All users that have access to customer accounts in Accounts also have permission to make updates or additions to Account records. Accounts records can contain general customer information, multiple address records for billing and shipping data, and comments. To collect more specific information, fullaccessusers can generate supplemental fields for capturing custom data. For more information, see Adding Custom Attributes.

A single Account record can be associated with one or more partner users. Associations are necessary if you plan to integrate Accounts with other areas of administration, such as configuration rules and commerce documents. To learn more about associations, see Defining Associations.

Accounts can also be integrated with commerce to auto-fill commerce document fields. Integrating Accounts with commerce speeds up document creation time by eliminating redundant manual data entry.

Managing User Accounts

The Accounts section maintains customer records. Using this page you can:

- Search for Customer Records
- Add Accounts Manually or through Integration
- Import Account Files
- Export Master Data and/or Address Information
- Create Custom Attributes for Accounts
- Customize Search: See Setup Account Search Criteria for more details.
- Customize Search Results: See Customizing Search Results for more details.

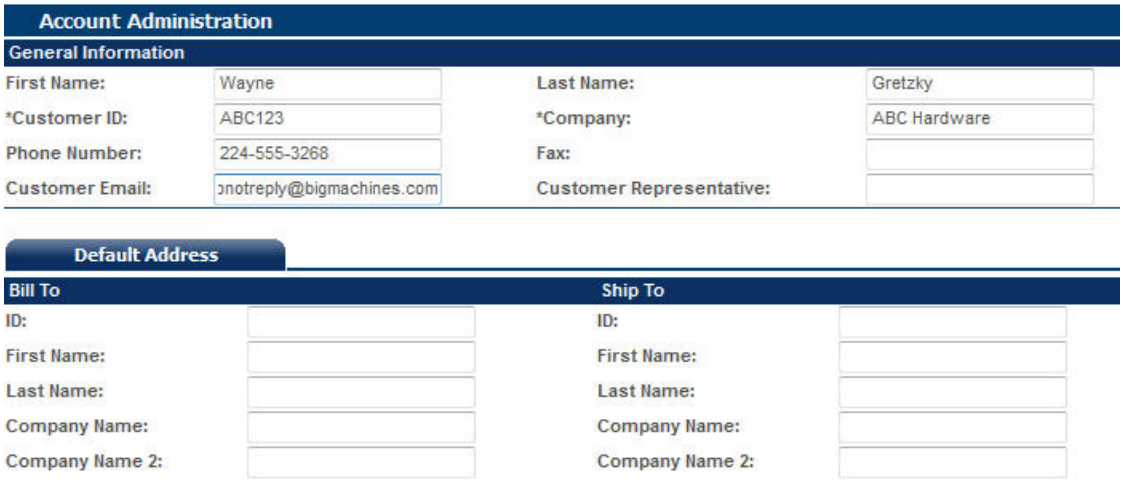
ADMINISTRATION

Creating an Account (Manually)

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Add** button.
3. Enter **General Account Information** and provide **Default Bill To/Ship To Addresses.**

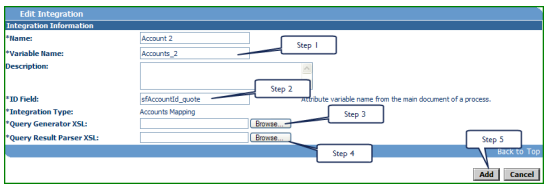
Customer ID and **Company Name** fields are required to create an Account.

4. In Additional Information, complete any custom attribute fields.
5. Click **Add**.



Creating an Account (Integration)

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click **Integration** to be taken to the Integration for CRM Page.
3. Click **Add**.
4. Enter a Name, Variable Name and Description for the new integration.
5. In the ID field, enter the attribute variable name from the main document of a process.
6. Click **Browse** to locate and add a Query Generator XSL file.
7. Click **Browse** to locate and add a Query Result Parser XSL file.
8. Click **Add** to add the new integration or click Cancel to go back to the page for managing integrations.

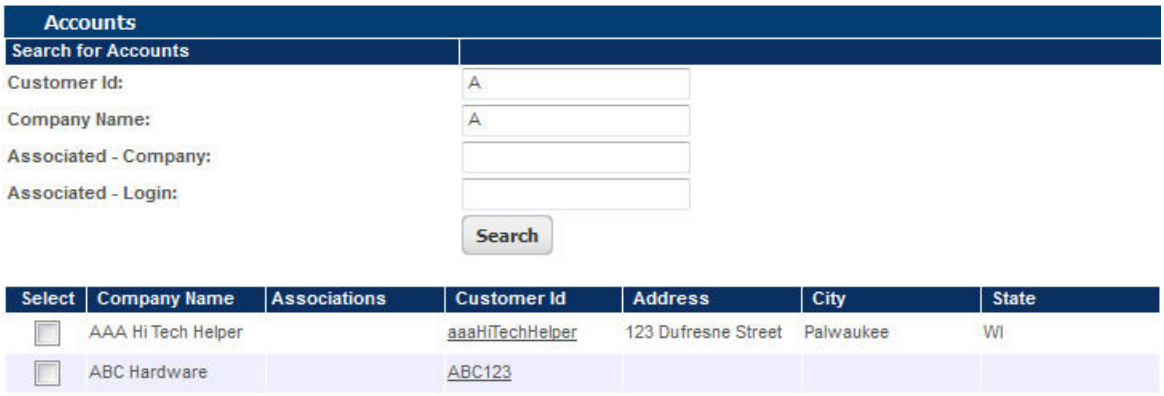


A Commerce Transaction ID field will appear after you've saved the Intgration.

9. Enter a commerce **Transaction ID** and click **Show Query Input XML** or Preview SOAP XML for additional integration details.

Managing Existing Accounts:

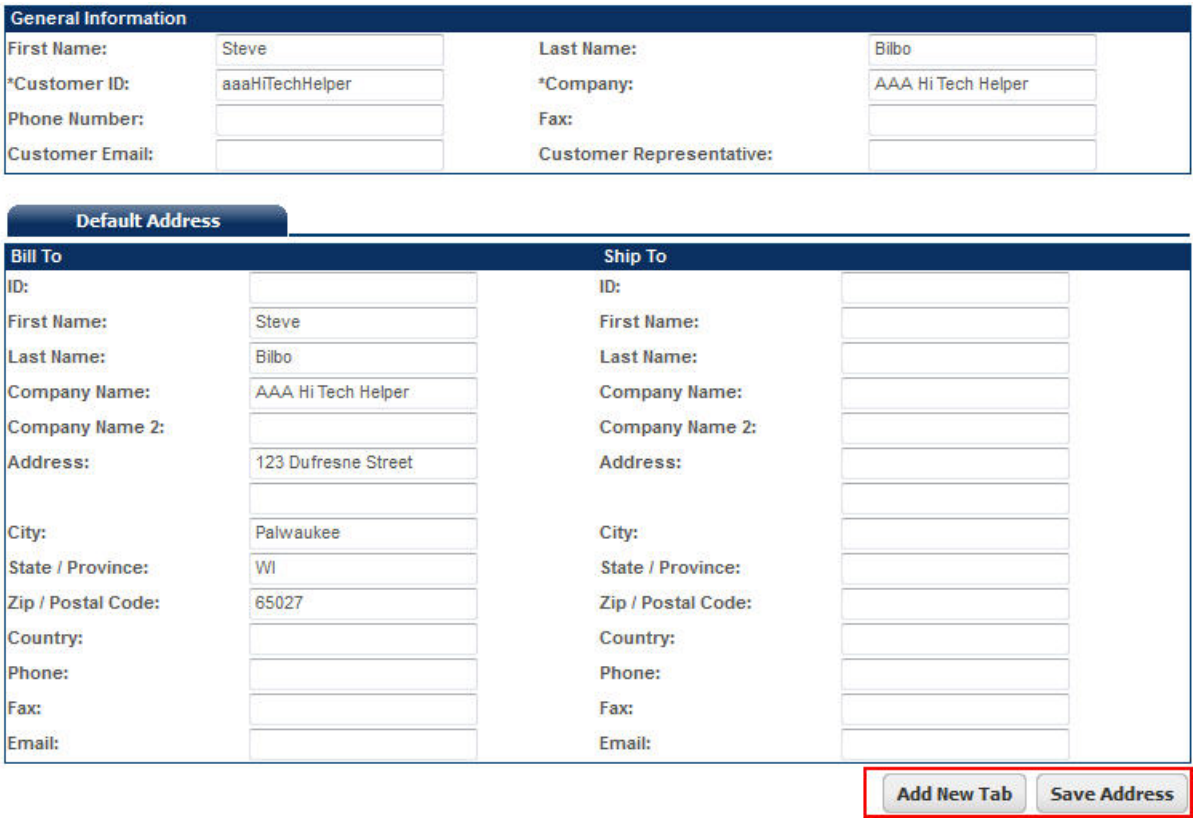
1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Search for the **Account** by Customer ID, Company Name, Associated-Company or Associated-Login.
3. Click **Search**.
4. Click on the **Customer ID**of the Account you want to edit.



5. Edit or enter account information in the fields provided.
6. Click **Add New Tab** to add an alternate address to new and existing Accounts records.
   - Enter a **tab label** for the new address, enter alternate address data, and click Save Address. Click on Delete Address when it is no longer required or valid.

     If you accidentally click the Add New Tab button instead of the Save Address button, cancel the operation. Otherwise, the new address record will be lost.

   - Alternate addresses appear as separate address tabs and can be viewed or modified by clicking the tab name.



7. Click on **Associations** to define account associations. See Defining Associations for more information.
8. Click **Update** to Save changes to information or click Back to return to the Accounts page without saving.

Importing Account Files

1. Prepare a .CSV file for Import.
2. Use the standard bulk upload actions to upload information.

> Click **Export Master Data** or **Export Address Data** for a template that you can use for the upload.

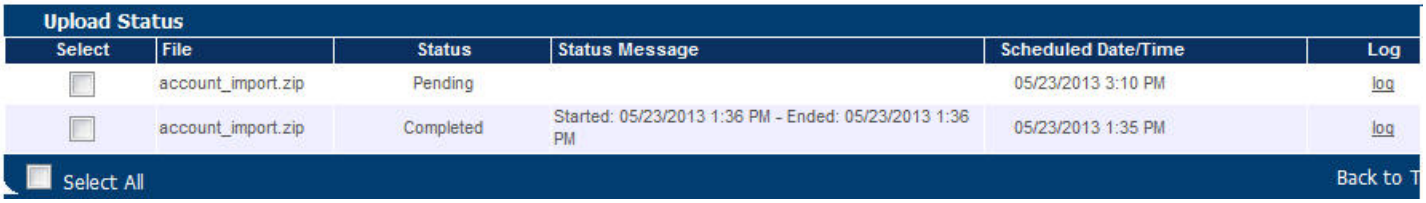
4. Enter information into the available fields. Customer ID (customer_id) and Company Name (company_name) are required.

	A	B	C	D	E	F	G	
1	Accounts Master							
2								
3	update	first_name	last_name	customer_id	company_name	customer_rep	phone	er
4	add	Steve	Bilbo	aaaHiTechHelper	AAA Hi Tech Helper			
5	add	Wayne	Gretzky	ABC123	ABC Hardware		224-555-3268	do
6	add	Ned	Finterwink	nedsComputerShed	Neds Computer Shed			
7	add	Beazus	Stankus	nerdherd	Nerd Herd			
8	add	Owen	Lawson	newOrleansDigital	New Orleans Digital			
9	add	Jason	OReilly	northeasternRack	Northeastern Rack			
10	add	Ron	Swanson	steelToolWerks	Steel Tool Werks			
11	add	Toni	Pastorino	ABC234	ABC Hardward - San Mateo		224-555-3269	do

5. Save the file and create a .zip file.
6. Navigate from Admin Home Page > Integration Platform > Accounts
7. Click the **Import** button.
8. Click the **Browse** button and choose a file.



9. Click **Add**. Repeat Steps 1-9 for all files to be imported.
10. Select the **file** and click the **Upload** button.
11. Click the **Refresh** button to see the status of the import.



12. Click the **log** to verify the upload has occurred without errors.

Export Master Data

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Export Master Data** button.

3. Select an option to Open or Save the file. The data will be downloaded as a .zip file.
4. Open the .zip file and double-click on the .csv file to open it.

	A	B	C	D	E	F	G	
1	Accounts Master							
2								
3	update	first_name	last_name	customer_id	company_name	customer_rep	phone	e
4	add	Steve	Bilbo	aaaHiTechHelper	AAA Hi Tech Helper			
5	add	Wayne	Gretzky	ABC123	ABC Hardware		224-555-3268	d
6	add	Ned	Finterwink	nedsComputerShed	Neds Computer Shed			
7	add	Beazus	Stankus	nerdherd	Nerd Herd			
8	add	Owen	Lawson	newOrleansDigital	New Orleans Digital			
9	add	Jason	OReilly	northeasternRack	Northeastern Rack			
10	add	Ron	Swanson	steelToolWerks	Steel Tool Werks			

Export Address Data

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the Export Address Data button.
3. Select an option to Open or Save the file. The data will be downloaded as a .zip file
4. Open the .zip file and double-click on the.csv file to open it.

	A	B	C	D	E	F	G	H	I	J	K	
1	Accounts A	ddresses										
2												
3	update	customer_	tab_label:	tab_label:	tab_label:	tab_label:	billing_firs	billing_last	billing_con	billing_con	billing_add	billi
4	add	aaaHiTech	Default Address				Steve	Bilbo	AAA Hi Tech Helper		123 Dufresne	S
5	add	ABC123	Default Address									
6	add	nedsComp	Default Address				Ned	Finterwink	Ned's Computer Shed		1938 SW Poto	
7	add	nerdherd	Default Address				Beazus	Stankus			7365 Swatmor	
8	add	newOrlear	Default Address				Owen	Lawson	New Orleans Digital S		54 Street Aven	
9	add	northeaste	Default Address				Jason	OReilly			93767 Floatilla	
10	add	steelToolV	Default Address				Ron	Swanson	Steel Tool Werks		570 Addison St	
11	add	steelToolV	Manufacturing Floor				Rick	Grieco	Steel Tool Werks		3744 Welles A	

NOTES



For Company Name and Customer ID search types, if you know only a partial company name or customer ID, you can search by the first few characters. The search engine locates all company names or customer IDs starting with the characters entered. Wild card searches are permitted for Company Name and Customer ID search types, however, they can be time consuming to complete.

- **Customer ID** - The Customer ID search option locates records based on the customer ID field required for each Accounts record. This type of search supports partial customer IDs with a trailing wildcard, meaning the search engine returns results based on a exact match between the search criteria entered and the beginning characters of the customer ID. A search using "123A" would return records with the following IDs: "123A23", "123ABG5" and "123A-345".
- **Company Name** - The Company Name search type locates records based on the company name field used to create an Accounts record. For flexibility, this search type returns results based on partial matches. If any part of a company name matches the criteria entered, the corresponding record is included in the search results set. For example, a search for "Acme" would result in records that have any of these names: "Acme Corp", "Big Acme Inc" and "Acme Corporation".

RELATED TOPICS

INTEGRATION GUIDE

OVERVIEW

The BigMachines application is capable of integrating with different types of applications. It does this using web services and with back-end systems, using Web Methods. There are two commonly used approaches to integration: Web Application Integration and Backend Systems Integration.

Integration Definitions

- Integration[1]
- FTP - File Transfer Protocol[2]
- Web Services[3]
- WSDL - Web Service Description Language[4]

ADMINISTRATION

Adding an Integration

1. Navigate from the Admin Home Page > Process Definition > Commerce Process > Integration XSL.
2. Click **Add**.
3. Select an **Integration Type**:
   - **Import:** Pulls data into BigMachines from a partner system. For example, in transactions, typically data that is imported into is accounts data from the partner site.

     Accounts mapping is a type of import integration used for the accounts section. It performs all the functions of a regular import integration operation.

   - **Export:** Pushes data from BigMachines into a partner system. As an example, export integrations are frequently to push quote PDFs to corresponding opportunities, every time there is a change made to the quote in .

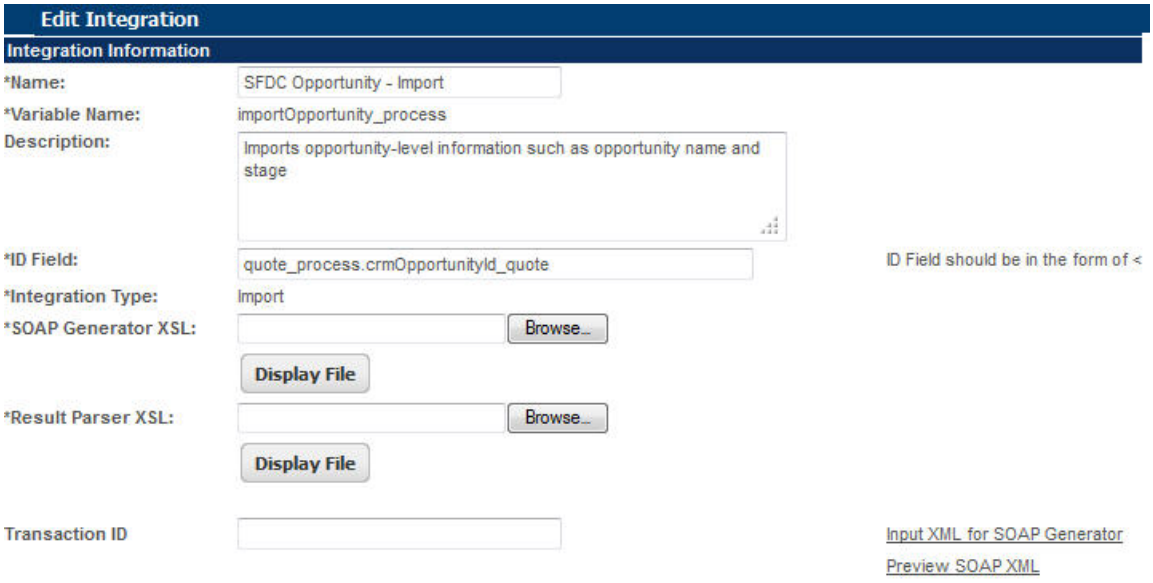
---

[1]The process of linking several applications to simply and automate business processes.
[2]FTP is a common way of transferring data. This is particularly userful for transferring large amount of data on a regular basis.
[3]A software system designed to support ineroperable machine-to-machine interaction of a network.
[4]Describes services as a collection of ports (network endpoints).

- **Delete :** Delete data on the partner system.
- **Query:**
- **Export Attachment:**

4. Click **Next**.
5. Fill out the Integration Information fields:

Field Name	Integration Type	Description
Name	All	Field that contains a name or label for the integration specified by the user.
Variable Name	All	Unique identifier that cannot by edited after creation.
Description	All	Detailed description field.
ID Field	All	Field that defines mapping between the partner entity and the entity. For example, suppose that the integration is being set up to import transaction data from the partner. The ID field must contain the variable name of the transaction field (for example, partner_opportunityid_field) into which the incomign data (for example, opportunity id) has to be populated.
SOAP Generator XSL	Import, Export, Delete, Query	Click **Browse** to select and load a SOAP Generator XSL based into the integration based on your use case.
Results Parser XSL	Import, Export, Delete	Click **Browse** to select and load a Results Parser XSL based into the integration based on your use case.
Transaction ID	Import, Export	Debugger field for transaction id.
Document	Export	Menu of main document of the process.
Document Number	Export	Debugger field for document number (typically line item document).



6. Click **Add**.

USE CASES

Use Case: Import Integration

Assume that your BigMachines instance is integrated with Salesforce.com and you are trying to integrate transaction data between the two.

1. Navigate from the Admin Home Page > Process Definition > Commerce Process > Documents > Main Document (Quote) Attributes.
2. Create an **attribute** to represent a Salesforce opportunity id field in the quote document with a variable name sfOpportunityId_quote. *Assume the main document (quote) is integrating with an opportunity in Salesforce.*
3. Navigate from the Admin Home Page > Process Definition > Commerce Process > Documents > Main Document (Quote) Actions.
4. Create an **action**, called **Save and Synch**,that will be used to trigger the integration.
5. Navigate from the Admin Home Page > Process Definition > Commerce Process > Integration XSL.
6. Click **Add**.
7. Select **Import (Query Generator and Result Parser)** then click **Next**.
8. Enter a **name**(*Import Opportunity*) and **variable name**(*imp_opp*) for the integration.
9. Enter the **variable name for the attribute you created in Step 2**, in the ID Field.



ID Field should be in the format: <document_var_name>.<att_var_name>. For example: *quote.sfOpportunityId_quote*.

10. Click **Browse** to select and load a SOAP Generator XSL based into the integration based on your use case.



The partner opportunity id is contained in the transaction's sfOpportunityId_quote field (ID field in the integration).

```
<?xml version="1.0" encoding="utf-8" ?>
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format"
  version="1.0">
  <xsl:output method="xml" />
  <xsl:variable name="main_doc" select="/transaction/data_xml/document[@document_id='4121558']" />
  <xsl:strip-space elements="*" />
- <xsl:template match="/">
    <!-- Begin SOAP XML -->
  - <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    - <soap:Header>
      - <QueryOptions xmlns="urn:partner.soap.sforce.com">
          <batchSize>1</batchSize>
        </QueryOptions>
      - <SessionHeader xmlns="urn:partner.soap.sforce.com">
        - <sessionId>
            <xsl:value-of select="/transaction/user_info/session_id" />
          </sessionId>
        </SessionHeader>
      </soap:Header>
    - <soap:Body>
      - <query xmlns="urn:partner.soap.sforce.com">
        - <queryString>
            select Id, Name, AccountId from Opportunity where Id='
            <xsl:value-of select="$main_doc/sfOpportunityId_quote" />
          </queryString>
        </query>
      </soap:Body>
    </soap:Envelope>
    <!-- End SOAP XML -->
  </xsl:template>
</xsl:stylesheet>
```

11. Click **Browse** to select and load a Result Parser XSL based into the integration based on your use case.



The result parser converts one XML format into another. If the partner system returns a valid id, then we populate the transaction fields projectname_quotewith the name value coming from the partner - sf:Name.

```
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sf2="urn:partner.soap.sforce.com"
    xmlns:sf="urn:sobject.partner.soap.sforce.com">
    <xsl:output method="xml" />
  - <xsl:template match="*">
    - <data_xml>
      - <xsl:if test="string-length(//sf:Id) > 0">
        - <document document_var_name="quote">
          - <projectname_quote>
              <xsl:value-of select="//sf:Name" />
            </projectname_quote>
          - <sfAccountId_quote>
              <xsl:value-of select="//sf:AccountId" />
            </sfAccountId_quote>
          </document>
        </xsl:if>
      </data_xml>
    </xsl:template>
  - <xsl:template name="formatDate">
      <xsl:param name="date" />
    - <xsl:if test="string-length($date) > 2">
        <xsl:value-of select="substring($date, 6,2)" />
        <xsl:text>/</xsl:text>
        <xsl:value-of select="substring($date, 9,2)" />
        <xsl:text>/</xsl:text>
        <xsl:value-of select="substring($date, 1,4)" />
      </xsl:if>
    </xsl:template>
  </xsl:stylesheet>
```

12. Click **Add**.



To view, you can click the Integration name. You can preview the SOAP XML by entering a commerce Transaction ID.

13. **Return to the commerce action**, Save and Synch, and click the Integration tab.
14. **Select Import Oppotunity** from the Integration List and move it to Selected Integration by clicking the right arrow. You can move it above or below other functions.
15. Click **Apply or Update** to save changes.
16. **Deploy the commerce process** and you are ready to trigger the integration for any transaction on the user-side.



Clicking the Save and Synch action on the transaction on the user side must retrieve the opportunity id from the partner and populate it into the sfOpportunityId field.

TROUBLESHOOTING

Debugging CRM Integration

- Updates may only be performed through the Integration API (via the CRM system). *If an error references permissions:*
    a. Log into the CRM system.
    b. Perform a manual update.



This error can occur when the administrator did not set up users' **Read/Write** access for a field.

- “Set as Primary” errors occur when there is an exception generated by the Apex code responsible for synching the primary quote with the Opportunity.
    - When this error occurs, it just appears as though the operation failed and nothing changed. *This is because the error is emailed to the installer of the package (who is not usually the user).* **Solution:** use an ALREADY primary quote. This causes the Apex exception to be displayed as an error in BigMachines when the user attempts to Save.
- Errors in synching the primary quote to the Opportunity are usually caused by the following:
    - A Quote product of the quote that will become the primary does not have the correct value in the BigMachines Part Number field. *This Product field links the Name field (included in query to create the Quote Product) to the Product ID (to create correct product on the Opportunity.*
        - If there are no Products that contain a BigMachines Part Number equal to a single Quote Product's Name field, the sync will fail and the following error message will be displayed: “one of the products does not have a price book entry."
    - A Quote product does not have a Price Book entry in the Opportunity Price Book (Standard Price Book for organizations that use a single Price Book).
        - If there is a Product with the BigMachines Part Number field that is equivalent to the Name field for every Quote Product, it is possible that the matching Product does not have a Price Book entry. This generates the following error message: “one of the products does not have a price book entry."
    - A Quote products does not have a Price Book entry in the Opportunity currency.
        - For multi-currency enabled organizations: A Product price must exist within the Opportunity Price Book in the Opportunity currency in order to be added to the Opportunity.
- When Save and Sync fails on the Create All (Products) step, the Quote may exist in Salesforce, but the Quote ID is not present on the BigMachines Quote.
    - The next time the user attempts to save the Quote, the following error message appears: "object already exists with external ID." *The uniqueness constraint is violated because the BigMachines Transaction ID is defined as a unique key*

within Salesforce Quotes; when the Quote ID field is blank in BigMachines, the query attempts to create a new object with the same ID. **Solution:** Invoke "Get ID" (this action is found in the Integration tab).

NOTES



There are four major criteria for performing an integration: size, frequency, format, and transmission synchronization. See Real Time Integration.



Attachments can only be exported to the Opportunity Object.

RELATED TOPICS

INTEGRATION GUIDE

OVERVIEW

The BigMachines application is capable of integrating with different types of applications. It does this using web services and with back-end systems, using Web Methods. There are two commonly used approaches to integration: Web Application Integration and Backend Systems Integration.

Integration Definitions

- Integration[1]
- FTP - File Transfer Protocol[2]
- Web Services[3]
- WSDL - Web Service Description Language[4]

ADMINISTRATION

Adding an Integration

1. Navigate from the Admin Home Page > Process Definition > Commerce Process > Integration XSL.
2. Click **Add**.
3. Select an **Integration Type**:
   - **Import:** Pulls data into BigMachines from a partner system. For example, in transactions, typically data that is imported into is accounts data from the partner site.

     Accounts mapping is a type of import integration used for the accounts section. It performs all the functions of a regular import integration operation.

   - **Export:** Pushes data from BigMachines into a partner system. As an example, export integrations are frequently to push quote PDFs to corresponding opportunities, every time there is a change made to the quote in .

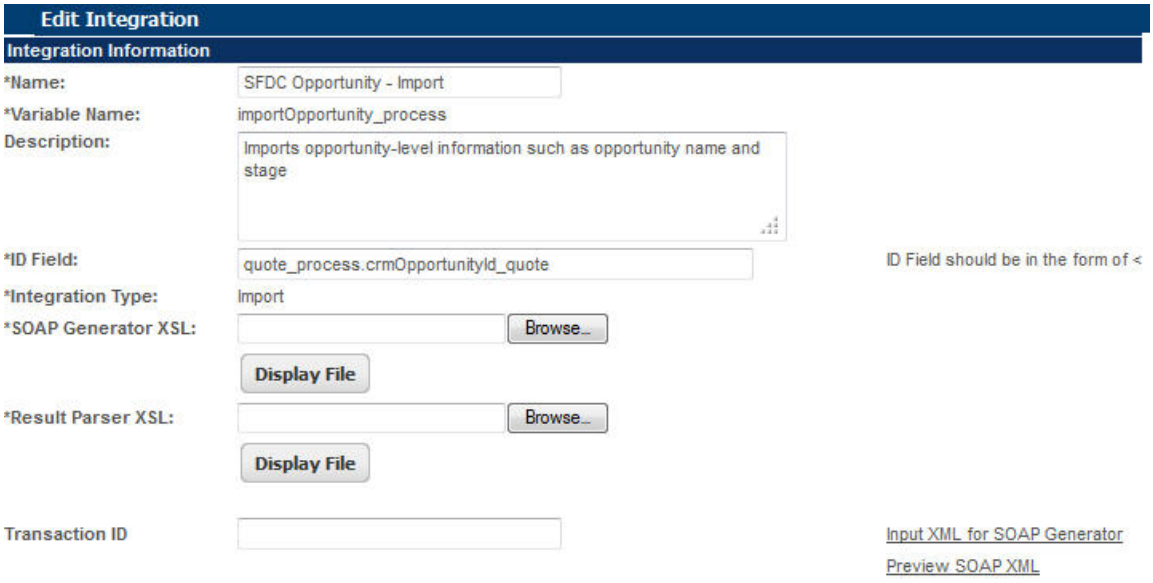
---

[1]The process of linking several applications to simply and automate business processes.
[2]FTP is a common way of transferring data. This is particularly userful for transferring large amount of data on a regular basis.
[3]A software system designed to support ineroperable machine-to-machine interaction of a network.
[4]Describes services as a collection of ports (network endpoints).

- **Delete :** Delete data on the partner system.
- **Query:**
- **Export Attachment:**

4. Click **Next**.
5. Fill out the Integration Information fields:

Field Name	Integration Type	Description
Name	All	Field that contains a name or label for the integration specified by the user.
Variable Name	All	Unique identifier that cannot by edited after creation.
Description	All	Detailed description field.
ID Field	All	Field that defines mapping between the partner entity and the entity. For example, suppose that the integration is being set up to import transaction data from the partner. The ID field must contain the variable name of the transaction field (for example, partner_opportunityid_field) into which the incomign data (for example, opportunity id) has to be populated.
SOAP Generator XSL	Import, Export, Delete, Query	Click **Browse** to select and load a SOAP Generator XSL based into the integration based on your use case.
Results Parser XSL	Import, Export, Delete	Click **Browse** to select and load a Results Parser XSL based into the integration based on your use case.
Transaction ID	Import, Export	Debugger field for transaction id.
Document	Export	Menu of main document of the process.
Document Number	Export	Debugger field for document number (typically line item document).



6. Click **Add**.

USE CASES

Use Case: Import Integration

Assume that your BigMachines instance is integrated with Salesforce.com and you are trying to integrate transaction data between the two.

1. Navigate from the Admin Home Page > Process Definition > Commerce Process > Documents > Main Document (Quote) Attributes.
2. Create an **attribute** to represent a Salesforce opportunity id field in the quote document with a variable name sfOpportunityId_quote. *Assume the main document (quote) is integrating with an opportunity in Salesforce.*
3. Navigate from the Admin Home Page > Process Definition > Commerce Process > Documents > Main Document (Quote) Actions.
4. Create an **action**, called **Save and Synch**,that will be used to trigger the integration.
5. Navigate from the Admin Home Page > Process Definition > Commerce Process > Integration XSL.
6. Click **Add**.
7. Select **Import (Query Generator and Result Parser)** then click **Next**.
8. Enter a **name**(*Import Opportunity*) and **variable name**(*imp_opp*) for the integration.
9. Enter the **variable name for the attribute you created in Step 2**, in the ID Field.



ID Field should be in the format: <document_var_name>.<att_var_name>. For example: *quote.sfOpportunityId_quote*.

10. Click **Browse** to select and load a SOAP Generator XSL based into the integration based on your use case.



The partner opportunity id is contained in the transaction's sfOpportunityId_quote field (ID field in the integration).

```
<?xml version="1.0" encoding="utf-8" ?>
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format"
  version="1.0">
  <xsl:output method="xml" />
  <xsl:variable name="main_doc" select="/transaction/data_xml/document[@document_id='4121558']" />
  <xsl:strip-space elements="*" />
- <xsl:template match="/">
    <!-- Begin SOAP XML -->
  - <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    - <soap:Header>
      - <QueryOptions xmlns="urn:partner.soap.sforce.com">
          <batchSize>1</batchSize>
        </QueryOptions>
      - <SessionHeader xmlns="urn:partner.soap.sforce.com">
        - <sessionId>
            <xsl:value-of select="/transaction/user_info/session_id" />
          </sessionId>
        </SessionHeader>
      </soap:Header>
    - <soap:Body>
      - <query xmlns="urn:partner.soap.sforce.com">
        - <queryString>
            select Id, Name, AccountId from Opportunity where Id='
            <xsl:value-of select="$main_doc/sfOpportunityId_quote" />
          </queryString>
        </query>
      </soap:Body>
    </soap:Envelope>
    <!-- End SOAP XML -->
  </xsl:template>
</xsl:stylesheet>
```

11. Click **Browse** to select and load a Result Parser XSL based into the integration based on your use case.



The result parser converts one XML format into another. If the partner system returns a valid id, then we populate the transaction fields projectname_quotewith the name value coming from the partner - sf:Name.

```
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sf2="urn:partner.soap.sforce.com"
    xmlns:sf="urn:sobject.partner.soap.sforce.com">
    <xsl:output method="xml" />
  - <xsl:template match="*">
    - <data_xml>
      - <xsl:if test="string-length(//sf:Id) > 0">
        - <document document_var_name="quote">
          - <projectname_quote>
              <xsl:value-of select="//sf:Name" />
            </projectname_quote>
          - <sfAccountId_quote>
              <xsl:value-of select="//sf:AccountId" />
            </sfAccountId_quote>
          </document>
        </xsl:if>
      </data_xml>
    </xsl:template>
  - <xsl:template name="formatDate">
      <xsl:param name="date" />
    - <xsl:if test="string-length($date) > 2">
        <xsl:value-of select="substring($date, 6,2)" />
        <xsl:text>/</xsl:text>
        <xsl:value-of select="substring($date, 9,2)" />
        <xsl:text>/</xsl:text>
        <xsl:value-of select="substring($date, 1,4)" />
      </xsl:if>
    </xsl:template>
  </xsl:stylesheet>
```

12. Click **Add**.



To view, you can click the Integration name. You can preview the SOAP XML by entering a commerce Transaction ID.

13. **Return to the commerce action**, Save and Synch, and click the Integration tab.
14. **Select Import Oppotunity** from the Integration List and move it to Selected Integration by clicking the right arrow. You can move it above or below other functions.
15. Click **Apply or Update** to save changes.
16. **Deploy the commerce process** and you are ready to trigger the integration for any transaction on the user-side.



Clicking the Save and Synch action on the transaction on the user side must retrieve the opportunity id from the partner and populate it into the sfOpportunityId field.

TROUBLESHOOTING

Debugging CRM Integration

- Updates may only be performed through the Integration API (via the CRM system). *If an error references permissions:*
    a. Log into the CRM system.
    b. Perform a manual update.



This error can occur when the administrator did not set up users' **Read/Write** access for a field.

- “Set as Primary” errors occur when there is an exception generated by the Apex code responsible for synching the primary quote with the Opportunity.
    - When this error occurs, it just appears as though the operation failed and nothing changed. *This is because the error is emailed to the installer of the package (who is not usually the user).* **Solution:** use an ALREADY primary quote. This causes the Apex exception to be displayed as an error in BigMachines when the user attempts to Save.
- Errors in synching the primary quote to the Opportunity are usually caused by the following:
    - A Quote product of the quote that will become the primary does not have the correct value in the BigMachines Part Number field. *This Product field links the Name field (included in query to create the Quote Product) to the Product ID (to create correct product on the Opportunity.*
        - If there are no Products that contain a BigMachines Part Number equal to a single Quote Product's Name field, the sync will fail and the following error message will be displayed: “one of the products does not have a price book entry."
    - A Quote product does not have a Price Book entry in the Opportunity Price Book (Standard Price Book for organizations that use a single Price Book).
        - If there is a Product with the BigMachines Part Number field that is equivalent to the Name field for every Quote Product, it is possible that the matching Product does not have a Price Book entry. This generates the following error message: “one of the products does not have a price book entry."
    - A Quote products does not have a Price Book entry in the Opportunity currency.
        - For multi-currency enabled organizations: A Product price must exist within the Opportunity Price Book in the Opportunity currency in order to be added to the Opportunity.
- When Save and Sync fails on the Create All (Products) step, the Quote may exist in Salesforce, but the Quote ID is not present on the BigMachines Quote.
    - The next time the user attempts to save the Quote, the following error message appears: "object already exists with external ID." *The uniqueness constraint is violated because the BigMachines Transaction ID is defined as a unique key*

within Salesforce Quotes; when the Quote ID field is blank in BigMachines, the query attempts to create a new object with the same ID. **Solution:** Invoke "Get ID" (this action is found in the Integration tab).

NOTES



There are four major criteria for performing an integration: size, frequency, format, and transmission synchronization. See Real Time Integration.



Attachments can only be exported to the Opportunity Object.

RELATED TOPICS

SALESFORCE INTEGRATION GUIDES

OVERVIEW

Salesforce Integration PDF Guides *(click to download)*
Salesforce.com Integration Guide v4.14
Salesforce.com Integration Guide v4.13
Salesforce.com Integration Guide v4.11
Salesforce.com Integration Guide v4.09
Salesforce.com Integration Guide v3.17
Salesforce.com User Management Implementation Guide

ADMINISTRATION

Mapping an Attribute to a SFDC Field

Within Salesforce

1. Create a new **Opportunity** field for the information to display.
2. Navigate from: Setup > Customize > Opportunities > Fields.
3. Click **New**. *Take note of the variable's API name (on the Opportunity Fields page).*

Within BigMachines

1. Locate the integration function that exports to Salesforce.
2. Navigate from: Admin > Process Definition > Integration XSL Files.
3. Select the name for this export integration (usually SFDC Quote - Upsert).
4. Click on **Display File** next to SOAP Generator XSL. *You'll need to save this file.*
5. Add the following code:

```
<SalesForceFieldName>
<xsl:value-of select="$main_doc/bigMachinesAttribute_quote" />
</SalesForceFieldName>
```

   - SalesForceFieldName is the API name for the field in SalesForce (usually in the form of BigMachines_FieldName_c)
   - bigMachinesAttribute_quote is the quote level attribute name

6. Save this file. Keep a backup of the old file.

7. In the integration XSL files page within BigMachines, click **Browse** next to SOAP Generator XSL and select the new file that you saved.
8. Click **Update**.
9. Deploy Commerce.



By running any action that runs the integration function, you will see the new field populate. Clicking "Save" on a quote will typically start the export.



This is a multi-step process that requires Admin privileges within Salesforce and BigMachines.

Salesforce Integration FAQs

How do I identify which integration version a site is using?

- **Version 1:**
  - Nothing will be on the Installed Packages page (but be aware as this may also be the case for Version 3).
  - No BigMachines Objects will be located on the Objects page.
- **Version 2:**
  - The Quote object was introduced in Version 2. Verify that you can create a quote.
    - The installed package will contain any of the following words: "BigMachines", "CPP", "LFE", etc. **OR**
    - Navigate from: Salesforce > Setup > Create > Objects. Look for a "BigMachines Quote" or "Quote" object.
- **Version 3:**
  - The "Set as Primary" button and the Quote Product object were introduced in Version 3.
    - Look for the "Set as Primary" button on the Quote related list. **OR**
    - Navigate from: Salesforce > Setup > Create > Objects and locate the Quote Primary Object.

- **Version 4:**
    - Navigate from: Salesforce > Setup > App Setup > View Installed Packages
        - Find an item called "BigMachines Commerce Integration". The version number should be 4.x. **OR**
        - Navigate from: All Tabs > BigMachines Site Setup tab. If you can locate this tab, you are using Version 4.x.

TROUBLESHOOTING

Using the Salesforce Debug Logs

1. Locate a Quote that has an issue.
2. Turn the **Debug Logs** on in Salesforce.
    a. Navigate from: Setup > Monitoring > Debug Logs > “New” button.
    b. Add it for the user you are logged in and/or the user that is performing the Saves to SFDC.
3. Attempt to save the Quote that is causing the issue.
4. Return to the **Debug Logs** and look for any logs that are listed as a “Failure” in the **Status** column.

Why can't I map the Quote Field to the Opportunity Field?

The Field may be set as a **Date/Time** type. Change the Field to **String** type.

Save and Synch action failed

This occurs when attempting to communicate with a partner site. When the action fails, the following error message appears: "sObject type 'BIGM_Quote__c' is not supported. If you are attempting to use a custom object, be sure to append the '__c' after the entity name. Please reference your WSDL or the describe call for the appropriate names."

Solution: Update the BigMachines custom quote object and update the field name to `BIGM_Quote`.



As a general rule, items added by the integration, fields, and objects should be prepended with `BIGM_`

Open Quote action failed

This occurs when attempting to communicate with a partner site. When the action fails, the following error message appears: No such column 'BIGM_Primary__c' on entity 'BIGM_Quote__c'. If you are attempting to use a custom field, be sure to append the '__c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names."

Solution: Update the primary field name to `BIGM_Primary`.

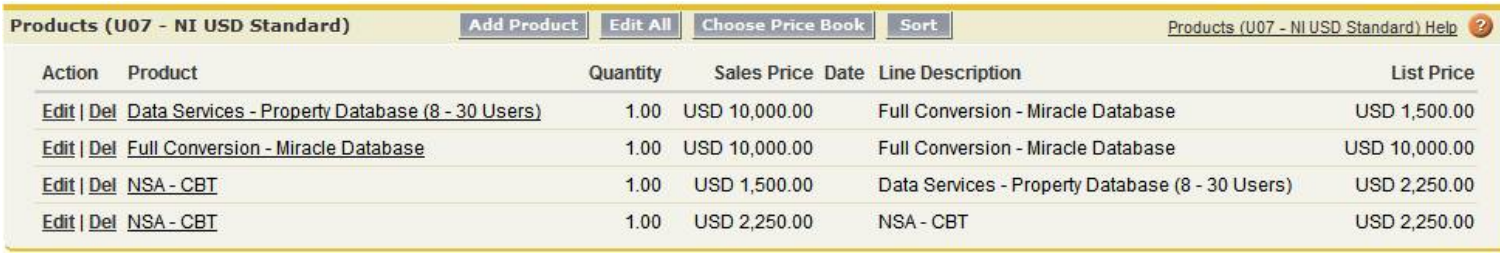
As a general rule, items added by the integration, fields, and objects should be prepended with `BIGM_`

Multiple instances of a single Quote are appearing in SFDC

Solution: When saving with SFDC, ensure that the necessary Integration XSLs are located on the action that is being performed.

Extra line item added to Opportunity Products section when a Primary Quote is Save & Synced

Problem: An extra line item is added to the Salesforce Opportunity Products section after performing a Save & Sync to a quote recently set as primary. *The rows appear inaccurately; a part may have another line item's list price and/or sales price.* The problem is that there are no Salesforce Parts for the BigMachines Models. When BigMachines passes the XML to Salesforce, the model is treated as a line item.



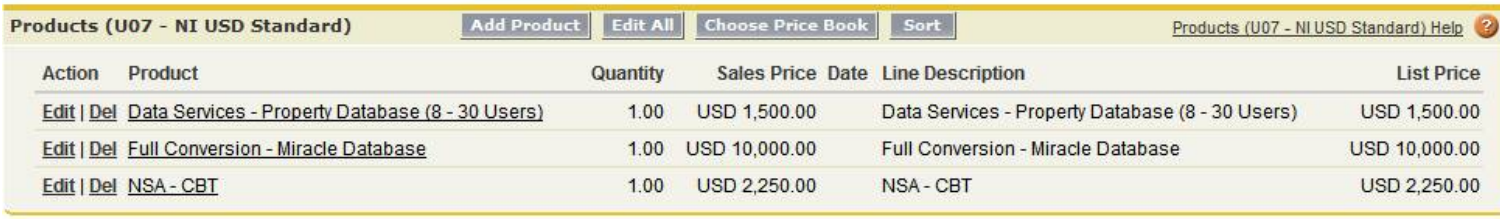
Products (U07 - NI USD Standard) | Add Product | Edit All | Choose Price Book | Sort | Products (U07 - NI USD Standard) Help

Action	Product	Quantity	Sales Price	Date	Line Description	List Price
Edit | Del	Data Services - Property Database (8 - 30 Users)	1.00	USD 10,000.00		Full Conversion - Miracle Database	USD 1,500.00
Edit | Del	Full Conversion - Miracle Database	1.00	USD 10,000.00		Full Conversion - Miracle Database	USD 10,000.00
Edit | Del	NSA - CBT	1.00	USD 1,500.00		Data Services - Property Database (8 - 30 Users)	USD 2,250.00
Edit | Del	NSA - CBT	1.00	USD 2,250.00		NSA - CBT	USD 2,250.00

Solution:

1. Update the **Save and Sync** and **Set as Primary Quote** Advanced Modification Rules.
2. When looping through the line items, add an argument to the "If" statement to ensure that the line item is model or part line item. **For example:**

```
if ( line.sfLineItemId_line == "" AND not(line.optional_line) AND len(line.sfPricebookEntryId_line) > 1) {
```



Products (U07 - NI USD Standard) | Add Product | Edit All | Choose Price Book | Sort | Products (U07 - NI USD Standard) Help

Action	Product	Quantity	Sales Price	Date	Line Description	List Price
Edit | Del	Data Services - Property Database (8 - 30 Users)	1.00	USD 1,500.00		Data Services - Property Database (8 - 30 Users)	USD 1,500.00
Edit | Del	Full Conversion - Miracle Database	1.00	USD 10,000.00		Full Conversion - Miracle Database	USD 10,000.00
Edit | Del	NSA - CBT	1.00	USD 2,250.00		NSA - CBT	USD 2,250.00

Selecting "New from Primary" after cloning a SFDC Opportunity failed

Problem: The following error messages display after a user attempts to select "New from Primary" after cloning a SFDC Opportunity:

```
Could not copy transaction.
CmBsData(-1) could not be found.
```

Solution: Ensure that the BigMachines Primary Quote attribute is located on the Opportunity. *The ID stored in this attribute is used to create a copy.*

Unable to clone a Quote

The inability to clone a Quote could be due to several factors. Look at the error messages to determine the cause:

- "Could not copy transaction. CmBsData(id = XXX) could not be found."
    - **Solution:** The quote with the id = XXX does not exist in BigMachines. This occurs if the BigMachines Quote was deleted. This error could also occur during a test instance of Salesforce where the BigMachines test instance may not contain the desired quote.
- "Could not perform Action: Save. Id does not exist."
    - **Solution:** Clear the BigMachines `sfLineItemIds_quote` field and click **Save**. *This occurs when some items in the `sfLineItemIds_quote` do not exist on the Salesforce Quote Object.*

Common Error Messages:

Invalid cross reference id.

Solution: Ensure all Parts actually exist in Salesforce and that the corresponding BigMachines Parts are utilizing the correct part IDs.

Partner Login Error for user (xxxx). An error occurred while communicating with the partner server at xxxx. This may have been due to invalide SOAP SXML generated by the XSL file.

Problem: This occurs when users attempt to sync their Partner login.

Solution: Ensure that users are using CTRL+V to paste their Security Token rather than using "right-click" paste.

Error in creating process: CmIntegration(id = 4129009) could not be found.

Problem: The following error message is received after **New Quote** is selected in Salesforce.

Solution: Ensure that the Account Integration XSL has been created and is mapped to the `_auto_refresh` button.

CmVersion(process_id = -1) could not be found.

Problem: The following error message is received after **Edit** is selected in Salesforce.

Solution: Ensure that the `process_id` is being passed to BigMachines. The URL parameter is `bm_cm_process_id`.

---

sObject type 'sObject' is not supported. If you are attempting to use a custom object, be sure to append the '__c' after the entity name. Please reference your WSDL or the describe call for the appropriate names.

Solution: Ensure that the **Export** call is being made with a sObject in it.

No such column 'BIGM_Old_Opportunity_Id__c' on entity 'Opportunity'. If you are attempting to use a custom field, be sure to append the '__c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names.

This error occurs when attempting to communicate with a partner site.

Solution: Update the BigMachines **Custom Opportunity** field. The **Field Name** should be `BIGM_Old_Opportunity_Id`.



As a general rule, items added by the integration, fields, and objects should be prepended with `BIGM_`

No such column 'AccountNumber' on entity 'Account'. If you are attempting to use a custom field, be sure to append the _ _c after the custom field name. Please reference your WSDL or the describe call for the appropriate names.

Check the import account query and import contact query files. The query string references several attributes that may not exist in the Salesforce account. The above error message occurs when these attributes do not exist and you are attempting to communicate with a partner site. **Solution:**

1. Remove the attribute name from the appropriate query XSL file.
2. Add the attribute into the Salesforce account.

Could not perform Action: Return to Opportunity

Problem: The Quote and the Opportunity were unable to be synced because at least one Product did not have a corresponding Price Book entry in the Opportunity's Price Book.

Solution: Manually add the Price Book entry in SFDC or contact a BigMachines administrator in order to run a Parts Sync. Ensure that the Model parent line item is either:

- Not being written back to Salesforce.
- Contains a Salesforce Part if it is being written back.

SFDC Integration Version 3 Issues:

promotePrimaryProdToOppty: execution of AfterUpdate

Problem: This occurs because BigMachinesQuoting.QuoteSyncException could not sync the Quote with the Opportunity. Price Book entries for all the products in the correct currency could not be found.

Solution: Ensure all products have a price in the Quote's currency.

SFDC hover box refreshes with a new page after selecting an action

Solution: Ensure your action JavaScript Redirect is working with `window.parent.location.href` **NOT** `window.location`.

Selecting SFDC hover box "Set as Primary" action takes you to an empty page

Solution: Revise the action JavaScript so the user returns to the **Opportunity** page using the below code rather than using `window.location`.

```
var targetURL = "/" + oppid;
```

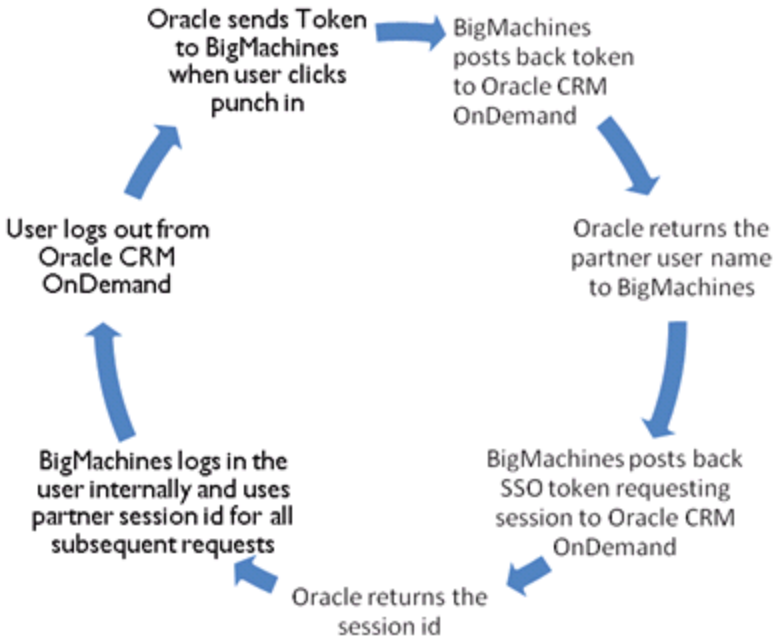
RELATED TOPICS

ORACLE CRM AUTHENTICATION

OVERVIEW

Oracle CRM and BigMachines are tightly integrated to provide a robust and secure user experience.

ORACLE - BigMachines Authentication Process



RELATED TOPICS

ORACLE FUSION INTEGRATION

OVERVIEW

This integration allows users to create quotes in BigMachines for a Fusion CRM opportunity. Users will be authenticated from BigMachines to Oracle Fusion via SSO.

- Leverage a link within Fusion CRM to create quotes in Big Machines
- Once created, the quote metadata is returned to Fusion CRM via a web service call
- View the quote details as part of the Fusion CRM opportunity
- Leverage a link within Fusion CRM to edit quotes in Big Machines
- Once edited, the updated quote metadata is returned to Fusion CRM via a web service call
- Quote proposal attachment can be included as a part of the custom quote
- Global logout implemented for all Oracle Fusion SSO functionality
- Standard authentication mechanism using Fusion-defined SAML approach
- Support different end points (URLs) for integration with different CRM objects (available at Integration XSL level)

SAML SSO Authentication

- Setup follows the standard Federated Authentication setup; for more information, see Single Sign-On (SSO).
    - An Oracle Fusion setup requires the addition of a RelayState URL Punch-In for the relevant quote following the standard Federated Authentication setup.
- Global logout is available and can be initiated from BigMachines and Fusion.
- Setup tested with Oracle Federation; Oracle's SAML server.

Endpoints

- Multiple endpoint capabilities:
    - A textbox to list the URL endpoint for each XSL.
        - Outbound URL for XSLs validated against a BigMachines-maintained whitelist.
- XSL endpoints are defined per integration XSL.
    - The user goes to the Edit Integration XSL page where the SOAP call may be specified.

ADMINISTRATION

Creating custom quote objects

1. Add the following fields:

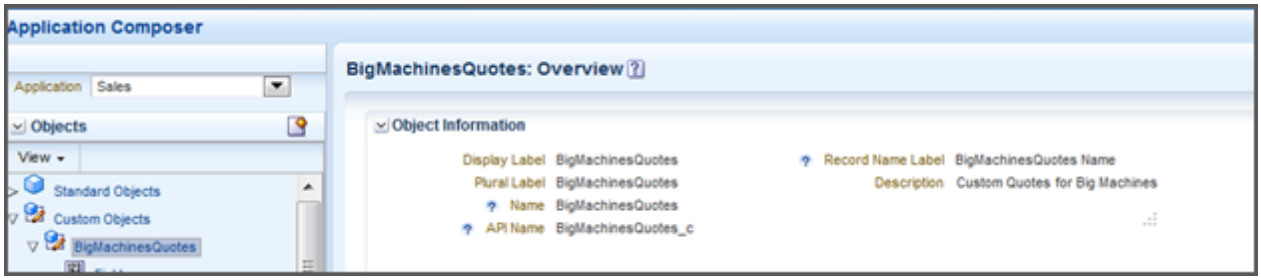
Field	Display Label	Type
QuoteId	Quote Id	Text
QuoteNr	Quote Nr	Text
Description	Description	Text
Status	Status	Text
Revenue	Revenue	String
Primary	Primary	Checkbox



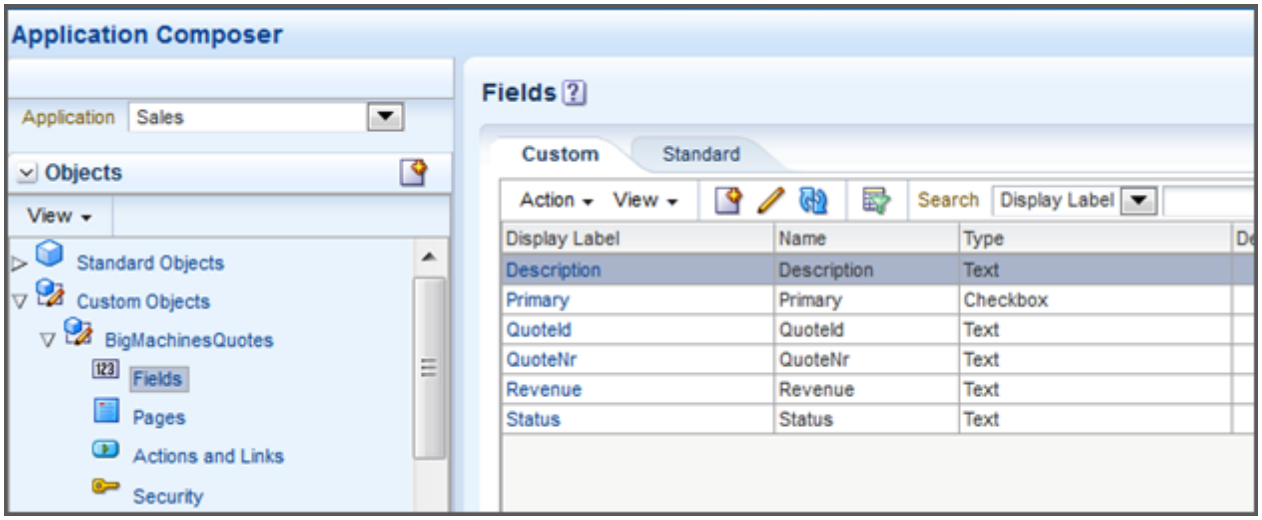
It is recommended that this be done within a Sandbox session; when satisfied with the results, publish the Sandbox to the mainline. *The custom object must be created outside of a Sandbox to enable testing.*

App Composer approach

1. Create a Custom object within the **Sales** Application.



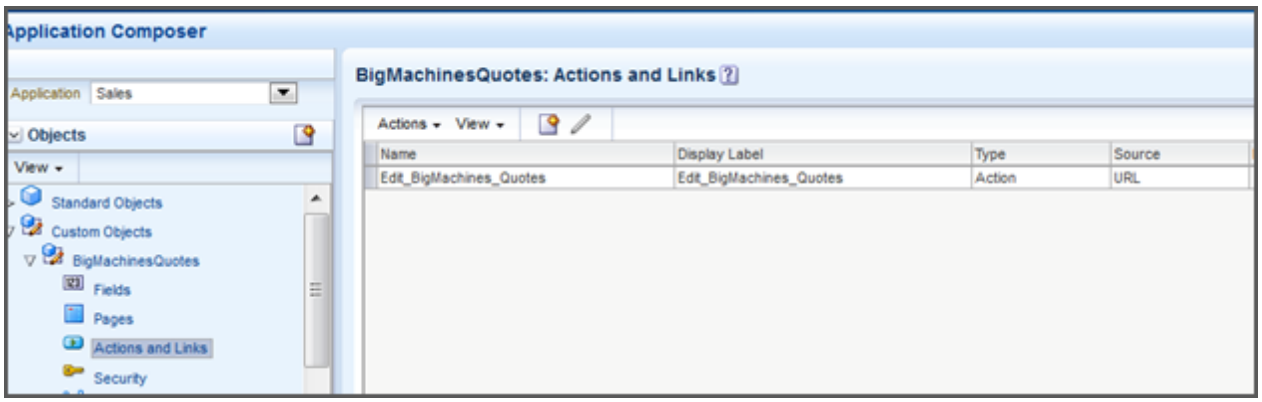
2. Create the following **Fields** in the **Custom** quote object:



3. Create a relationship between the **Opportunity** and **Custom** quote.



4. **Optional:** Enter a description.
5. Click **Save & Close**.
6. Create a link to edit the custom quote.

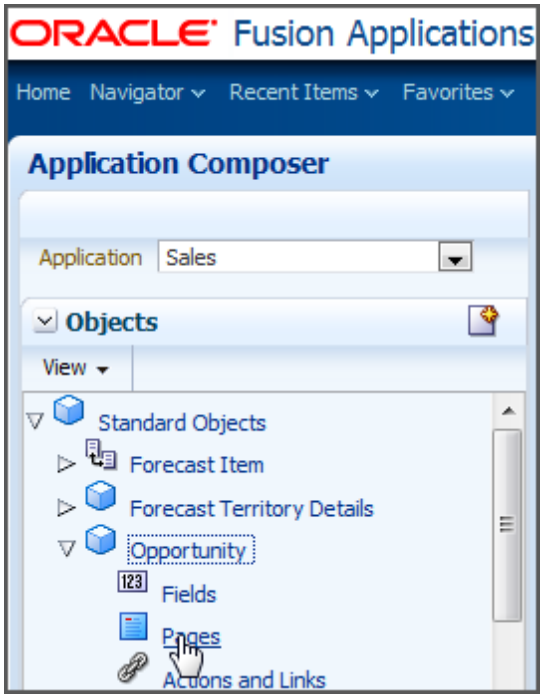


7. Create a link so the custom quote can be produced in BigMachines.

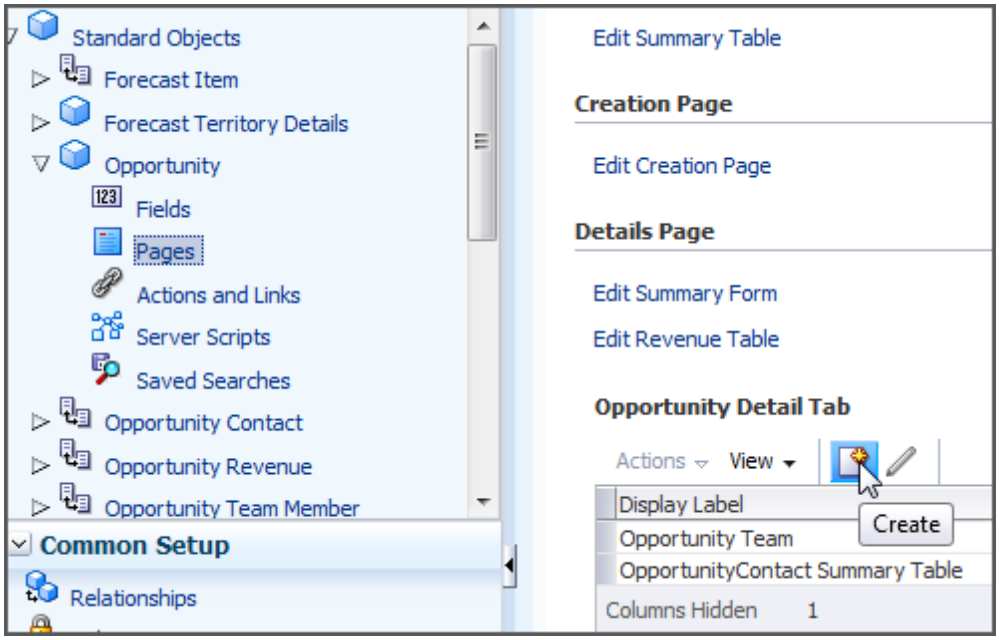


Adding a Tab to the Opportunity details (custom quote processing)

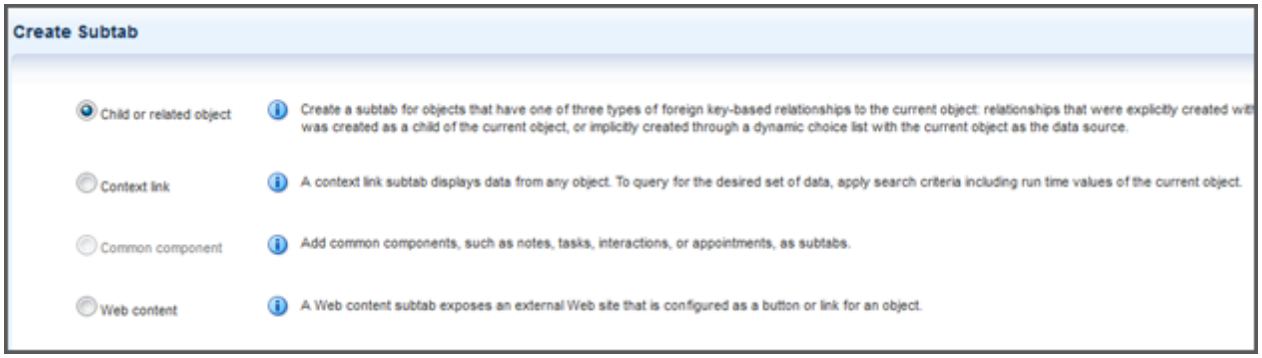
1. Click **Pages** link within the **Sales** Application.



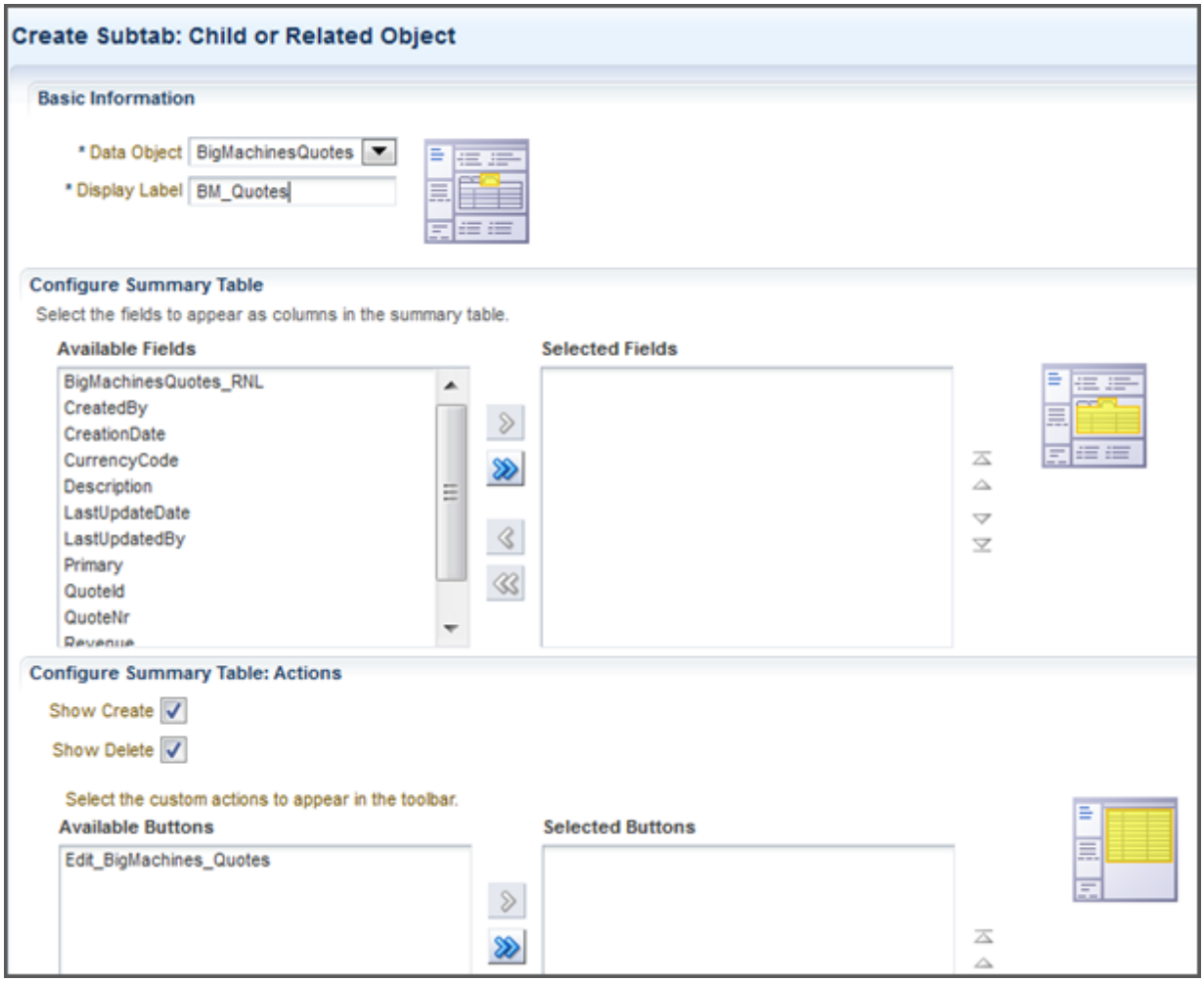
2. Click **Create** to add a new tab.



3. Click **Child or related object** when creating the subtab.



4. Click **Next**.
5. Select the custom quotes from the **Data Object**. *This refreshes the page with details (located at the bottom).*
6. Enter a valid **Display Label**.



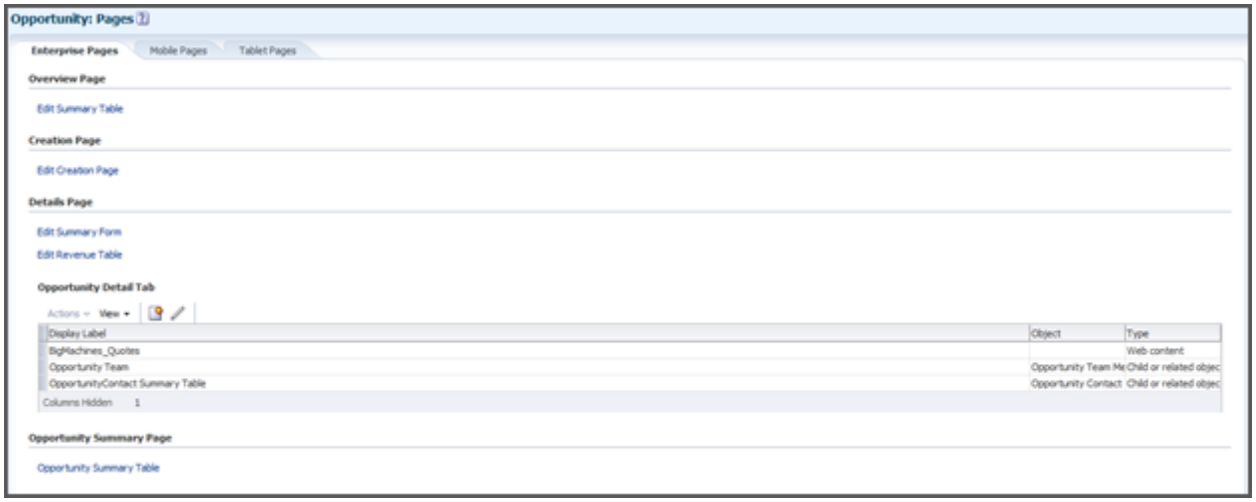
7. Select and move the **Available Fields** that you would like to appear as columns in the summary table to the **Selected Fields** area within **Configure Summary Table**.

8. Select and move the **Available Buttons** that you would like to appear as custom actions in the toolbar to the **Selected Buttons** area within **Configure Summary Table: Actions**.

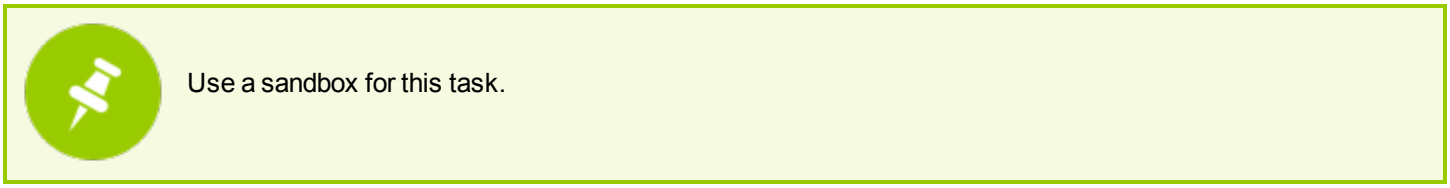
9. If you would like to disable the edit options, deselect the **Show Create** and **Show Delete** checkboxes.

10. Repeat steps 7 & 8 to make similar changes to the **Configure Detail Form** section.

11. Click **Save & Close** to create the subtab.



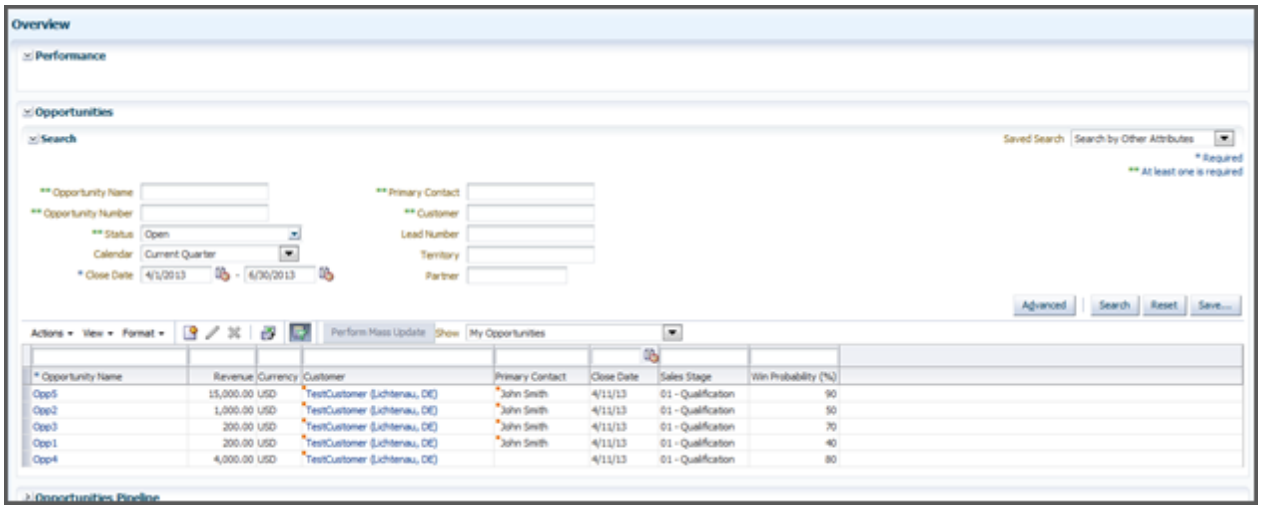
12. Click **Save & Close**.



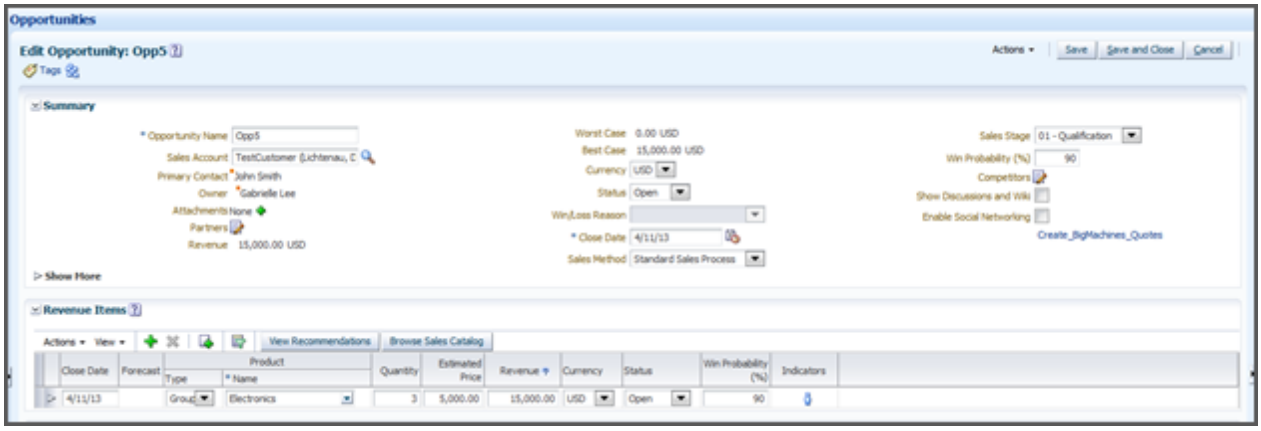
Use a sandbox for this task.

Validating the newly created tab

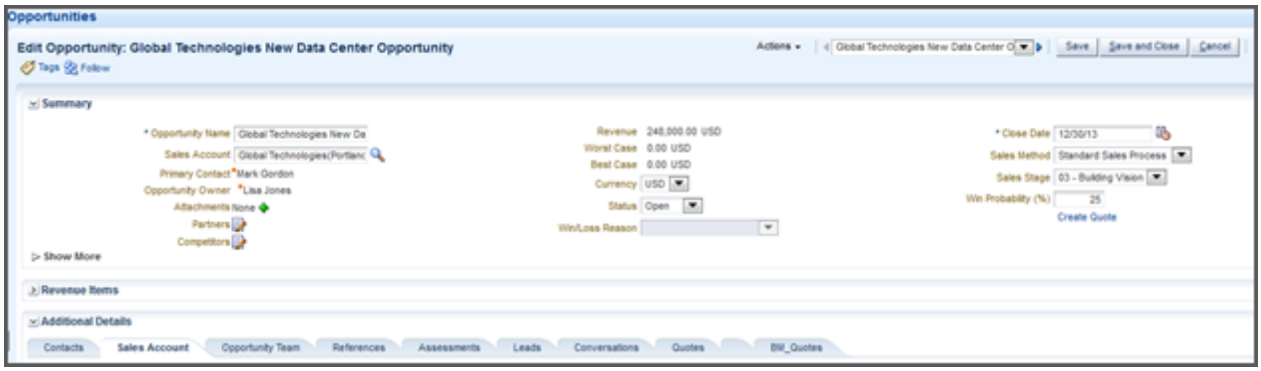
1. Login as a sales rep user. *Use the same sandbox that the tab was created in.*
2. Edit an **Opportunity**.



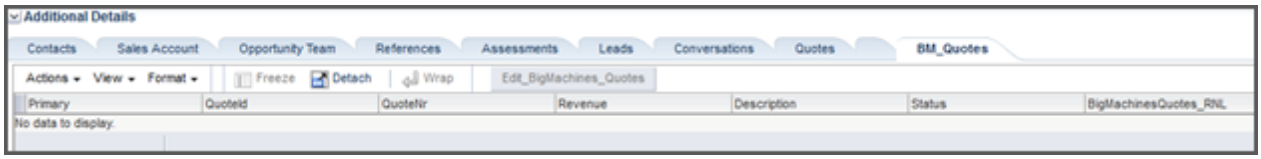
3. Select an **Opportunity**.



4. Expand the **Additional Details** section to view the custom quotes in a subtab.



5. Click a tab (e.g. BM_Quotes) to view quote details and the Opportunity custom link to "Create Quote".

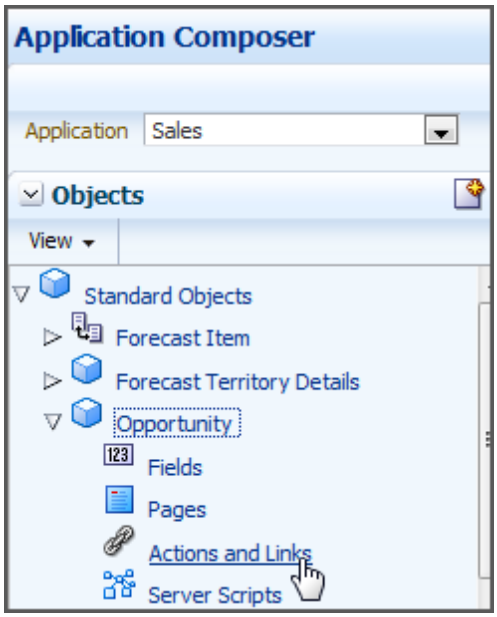


Creating the link

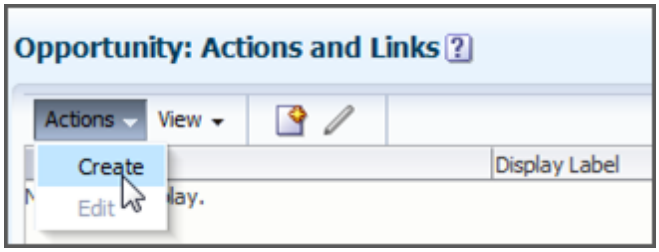
Once a quote is created in BigMachines, a web service call is made from Fusion CRM to create a custom quote. *The opportunity ID is required for this (known as **OptyId** within Fusion CRM).* The URL used to call BigMachines in order to create a quote, includes a **partnerAccountId**, which is known as **TargetPartyId** in Fusion CRM. The payload returned from BigMachines includes a QuoteId field (this is the internal QuoteId from within Fusion CRM). This field will be used by Fusion CRM when calling BigMachines to edit a quote.

Once a quote is updated in BigMachines, a web service call is made to Fusion CRM to update the relevant custom quote. The custom quote ID is required to identify the correct instance. This ID is a response to the createQuotey request, so BigMachines could store this. Otherwise, Fusion CRM could pass this as a part of the URL used to call BigMachines and edit a quote.

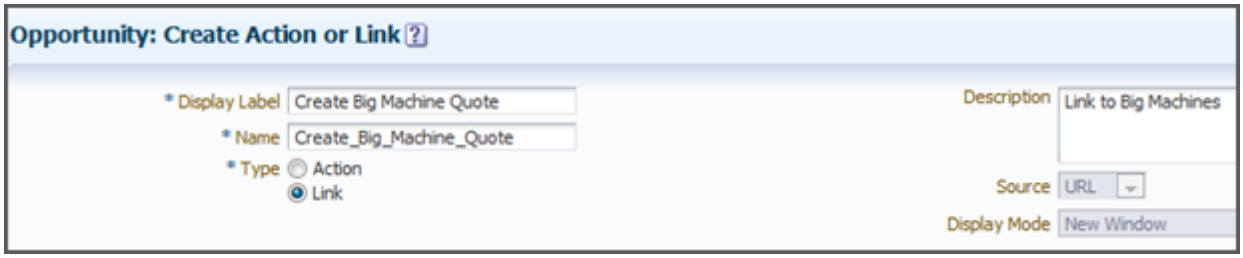
1. Choose **Actions and Links** beneath **Opportunity** within the **Sales** Application drop-down menu.



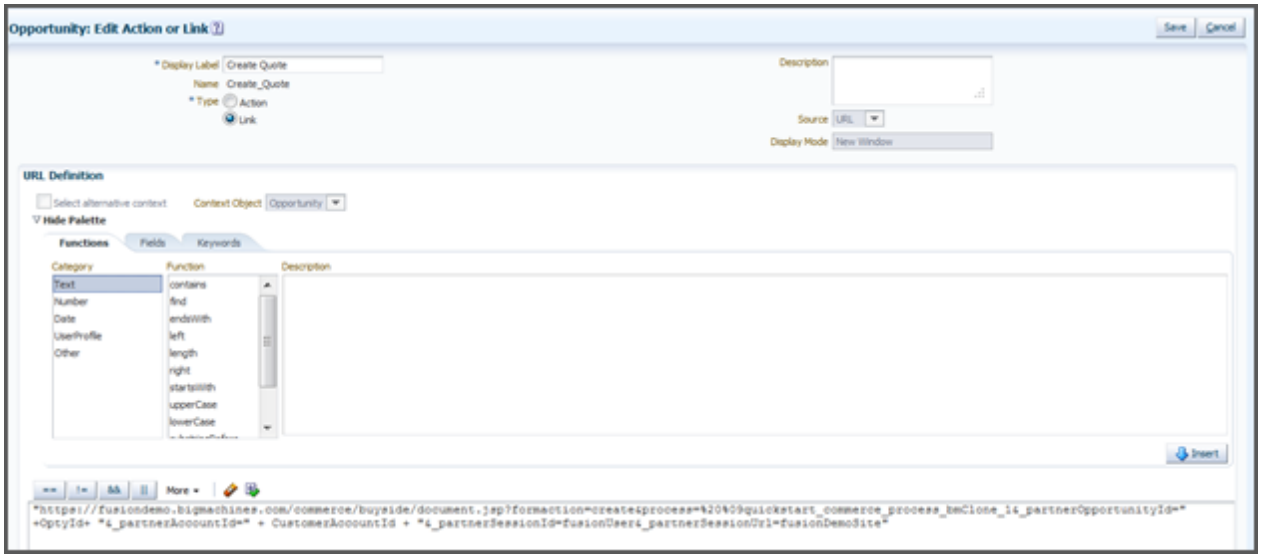
2. Choose **Create** from the **Actions** drop-down menu.



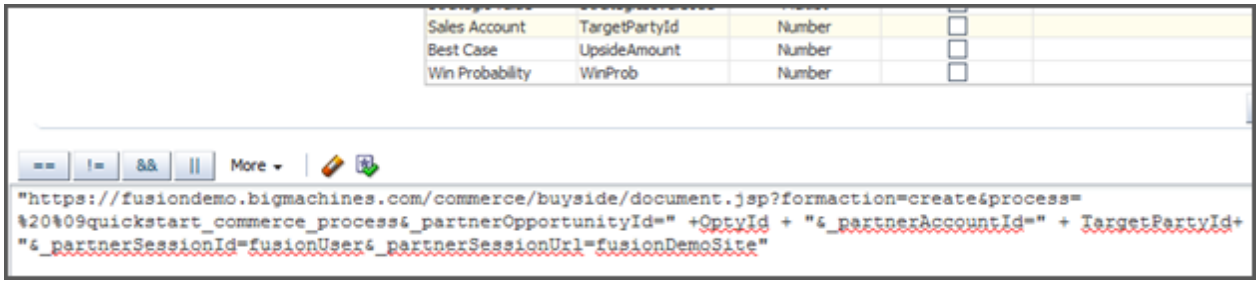
3. Enter the **Display Label**, **Name**, and **Type**.



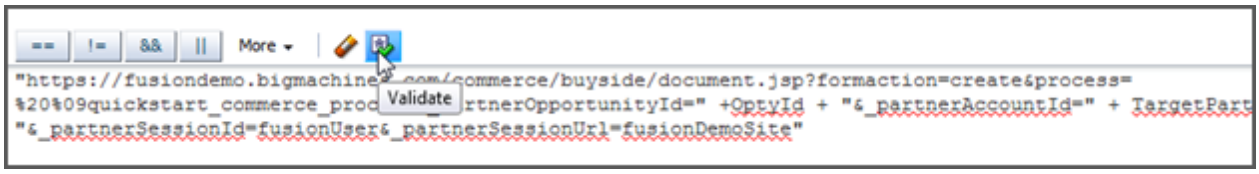
4. Select **OptyId** and click **Insert**.



5. Select **TargetPartyId** and click **Insert**.

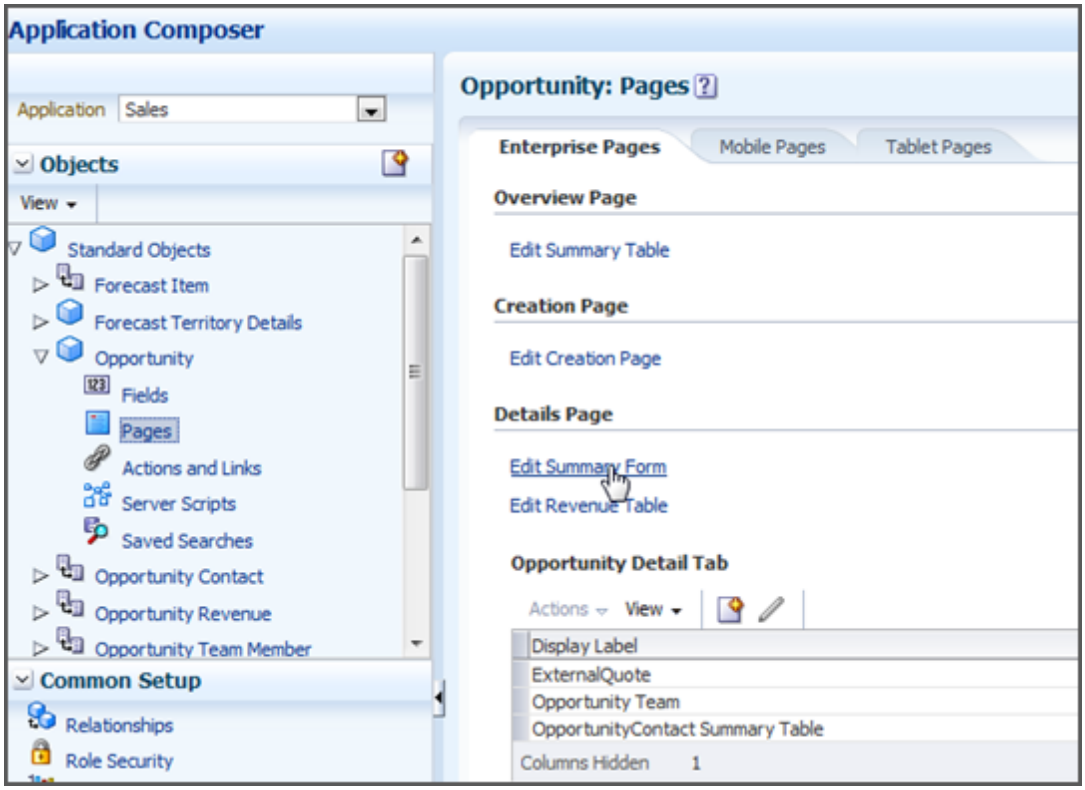


6. Click **Validate**.

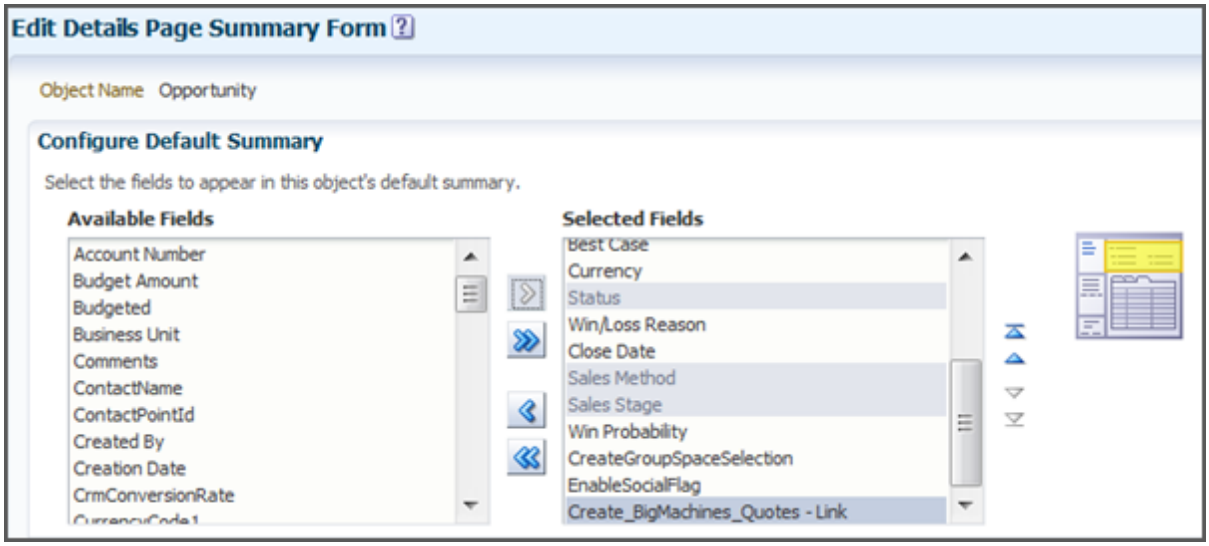


7. Select **Pages** beneath **Opportunity** within the **Sales** Application drop-down menu.

8. Click **Edit Summary Form** under the **Details Page** section.

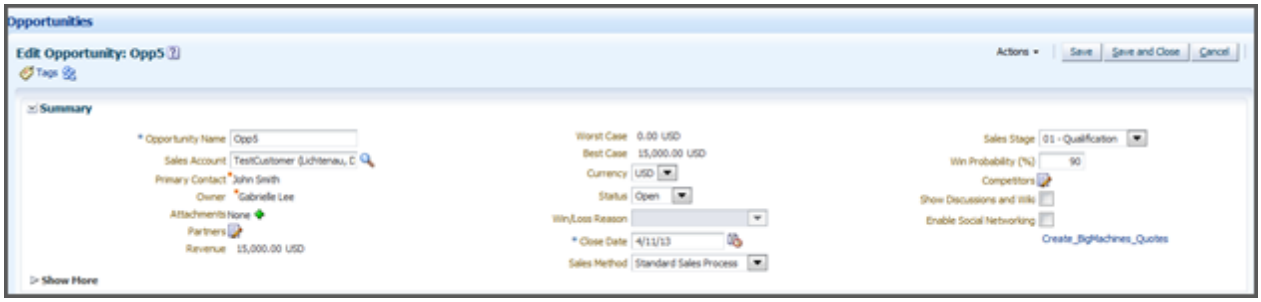


9. Add the link to the **Selected Fields** section so that it may appear on your page.



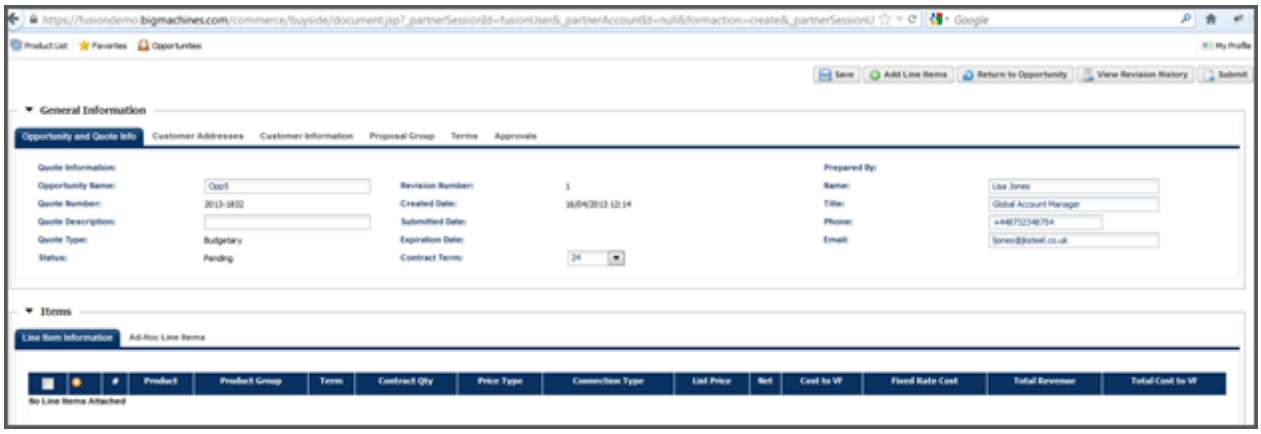
10. Click **Save and Close**.

11. Click on the link to test it.

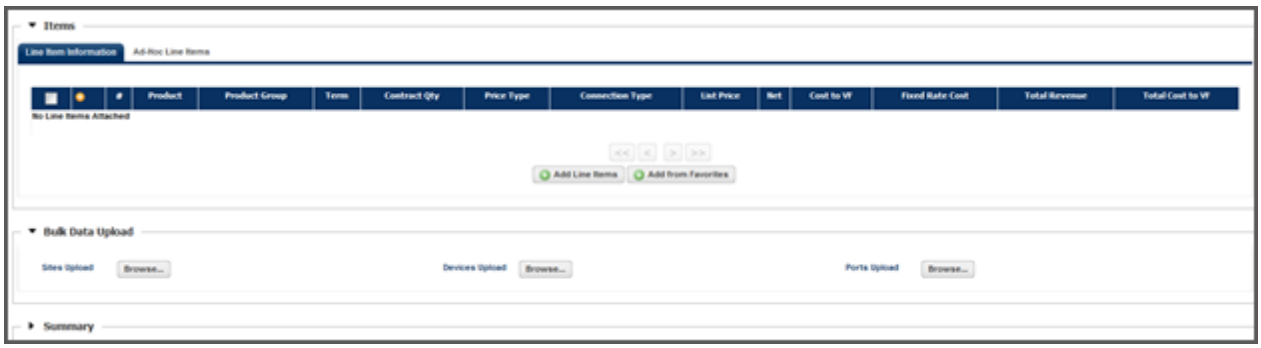


Testing the UI Integration

1. Log in to Fusion CRM as a sales rep.
2. Select an existing **Opportunity** OR create a new **Opportunity** to edit.
3. Click a custom link within the **Edit Opportunity** page summary section. *This opens a new BigMachines application where you can utilize CPQ functionality.*



4. Click **Add Line Item** to browse through the BigMachines product catalog.



Web Services Integration: Create new custom quotes

1. Create a **Custom quote** using the **OptyId** from the response for **Create Opportunity**. *Request example:*

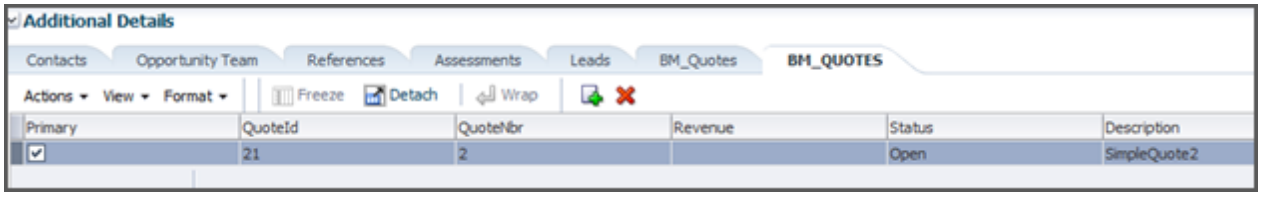
```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns5:createEntity xmlns:ns2="http://xmlns.oracle.com/adf/svc/types/" xmlns:n-
s3="http://xmlns.oracle.com/adf/svc/errors/" xmlns:ns4="commonj.sdo" xmlns:n-
s5="http://xmlns.oracle.com/apps/sales/custExtn/extnService/types/"
xmlns:ns6="/oracle/apps/sales/custExtn/views/common/">
<ns5:object xsi:type="ns6:BMquotes_c" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<ns6:RecordName>BMQ2</ns6:RecordName>
<ns6:QuoteId_c>21</ns6:QuoteId_c>
<ns6:QuoteNbr_c>2</ns6:QuoteNbr_c>
<ns6:Description_c>SimpleQuote2</ns6:Description_c>
<ns6:Status_c>Open</ns6:Status_c>
<ns6:Opportunity_Id_Opty2BMquotesRS>300000000703219</ns6:Opportunity_Id_Opty2BMquotesRS>
</ns5:object>
<ns5:objectName>BMquotes_c</ns5:objectName>
</ns5:createEntity>
</S:Body>
</S:Envelope>
```

Response example:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:-
typ="http://xmlns.oracle.com/apps/sales/custExtn/extnService/types/">
<env:Header/>
<env:Body>
<ns0:createEntityResponse xmlns:ns0="http://xmlns.oracle.com/apps/sales/custExtn/extnService/types/">
<ns2:result xsi:type="ns1:BMquotes_c" xmlns:ns2="http://xmlns.oracle.com/apps/sales/custExtn/extnService/types/"
xmlns:ns1="/oracle/apps/sales/custExtn/views/common/" xmlns:ns0="http://xmlns.oracle.com/adf/svc/types/" xmlns:x-
si="http://www.w3.org/2001/XMLSchema-instance">
<ns1:Id>300000000703580</ns1:Id>
<ns1:RecordName>BMQ2</ns1:RecordName>
<ns1:CreatedBy>integrationAccount1</ns1:CreatedBy>
<ns1:CreationDate>2013-08-14</ns1:CreationDate>
<ns1:LastUpdatedBy>integrationAccount1</ns1:LastUpdatedBy>
<ns1:LastUpdateDate>2013-08-14</ns1:LastUpdateDate>
<ns1:ObjectVersionNumber>1</ns1:ObjectVersionNumber>
<ns1:LastUpdateLogin xsi:nil="true"/>
<ns1:CpdrfVerSor xsi:nil="true"/>
<ns1:ConflictId>0</ns1:ConflictId>
<ns1:UserLastUpdateDate xsi:nil="true"/>
<ns1:CpdrfLastUpd xsi:nil="true"/>
<ns1:CpdrfVerPillar xsi:nil="true"/>
```

```
<ns1:JobDefinitionPackage xsi:nil="true"/>
<ns1:RequestId xsi:nil="true"/>
<ns1:JobDefinitionName xsi:nil="true"/>
<ns1:CurrencyCode>USD</ns1:CurrencyCode>
<ns1:CurcyConvRateType>Corporate</ns1:CurcyConvRateType>
<ns1:CorpCurrencyCode>USD</ns1:CorpCurrencyCode>
<ns1:SourceType>BMQUOTES_C</ns1:SourceType>
<ns1:AttachmentEntityName>BMquotes_c</ns1:AttachmentEntityName>
<ns1:PKProxy>300000000703580</ns1:PKProxy>
<ns1:IsOwner>true</ns1:IsOwner>
<ns1:QuoteId_c>21</ns1:QuoteId_c>
<ns1:QuoteNbr_c>2</ns1:QuoteNbr_c>
<ns1:Description_c>SimpleQuote2</ns1:Description_c>
<ns1:Primary_c>true</ns1:Primary_c>
<ns1:Revenue_c xsi:nil="true"/>
<ns1:Status_c>Open</ns1:Status_c>
<ns1:Opportunity_Id_Opty2BMquotesRS>300000000703219</ns1:Opportunity_Id_Opty2BMquotesRS>
</ns2:result>
</ns0:createEntityResponse>
</env:Body>
</env:Envelope>
```


2. Validate in the CRM.



NOTES

Proxy logged-in admins will be returned to the Users list when click Proxy Logout is clicked. *This occurs regardless of SSO logout URL specifications.*



Fusion package configuration will require additional Oracle documentation during setup. *Like the Oracle On Demand integration, some of the setup steps on Fusion are manual.*

RELATED TOPICS

REAL-TIME INTEGRATION

OVERVIEW

Real-time integration is the mechanism that sends transaction XML to a 3rd party server when a transition rule is fired. This is the most standard, out-of-the-box BigMachines functionality for implementing integration. Please take a look in the Related Pages section and click on the article "Standard XML Output" for the basic XML structure. See attachments for a Real-Time Integration Sample Document XML.

Integration Criteria	Value
Size	Large
Frequency	Triggered during workflow
Format	XML
Transmission Synchronization	Synchronous, however BigMachines can't parse the result.

ADMINISTRATION

Setting Up Real-Time Integration

1. Navigate from Admin Home Page > Integration > Integration Settings
2. Enter the **URL address** of the Host that has to be accessed.
3. Enter the **user name** of the partner server.



This is only required when posting to a site that requires a secure login.

4. Enter the **password** required to access the partner server. *Only required when posting to a site that requires a secure login.
5. Determine if you would like to enable authorization during the HTTP Post connection method by selecting:
    - **Yes:** When using Basic Authorization, the destination system can determine whether the incoming message is valid or not, without processing the entire request. As a result, fullaccessusers of the destination system have more control agains denial of service and/or security attacks.
    - **No:** A protocol that is unique to BigMachines. This works well for BigMachines - WebMethods integrations. Note: The

'No' setting will be set by default when customers upgrade their application to BigMachines v12.2.



6. Specify whether or not to integrate Users by selecting Yes or No.
7. Enter the **connection timeout** (in seconds). This indicates the duration after which the connection will be closed.
8. Enter the **maximum number of retries** that are allowed.
9. Specify the **Connection Method** by selecting either Web Methods or Http Post.
10. Click **Test Connection**. An error will be thrown if the connection cannot be established. For example, if you use the wrong user name, it will tell you that you entered the incorrect user name.
11. Click **Accept**t to accept the information and return to the Admin Home Page or click Cancel to return to the Admin Home Page without saving changes.

NOTES

All fields should be filled in to enable two-way communication between your partner server and the BigMachines trading hub.

When using the "HTTP POST" integration method, BigMachines will attempt to send data to the URL that you specify. To confirm reception of the data, the receiving URL must respond with the word 'success', all lower case and without the quotes, and nothing else. If BigMachines does not receive this response quickly enough, it will attempt to re-send the data, up to the number of retries specified in the integration settings. If any response other than 'success' is received, it will be considered as as failure and the integration data will be sent again. In most scenarios, it would be ideal to respond first, then process the data.

RELATED TOPICS

WEB APPLICATIONS INTEGRATION

OVERVIEW

Web applications integration is achieved using Simple Object Access Protocol (SOAP) messages. SOAP messages are exchanged between the partner and resulting in integration with partners like Salesforce and Siebel.

To be integrated with a partner, the system fullaccessuser must turn on the integration for that particular partner. Note: Partners change server URL sometimes, so if your integration fails during an upgrade, please verify the integration server settings with the system fullaccessuser.

ADMINISTRATION

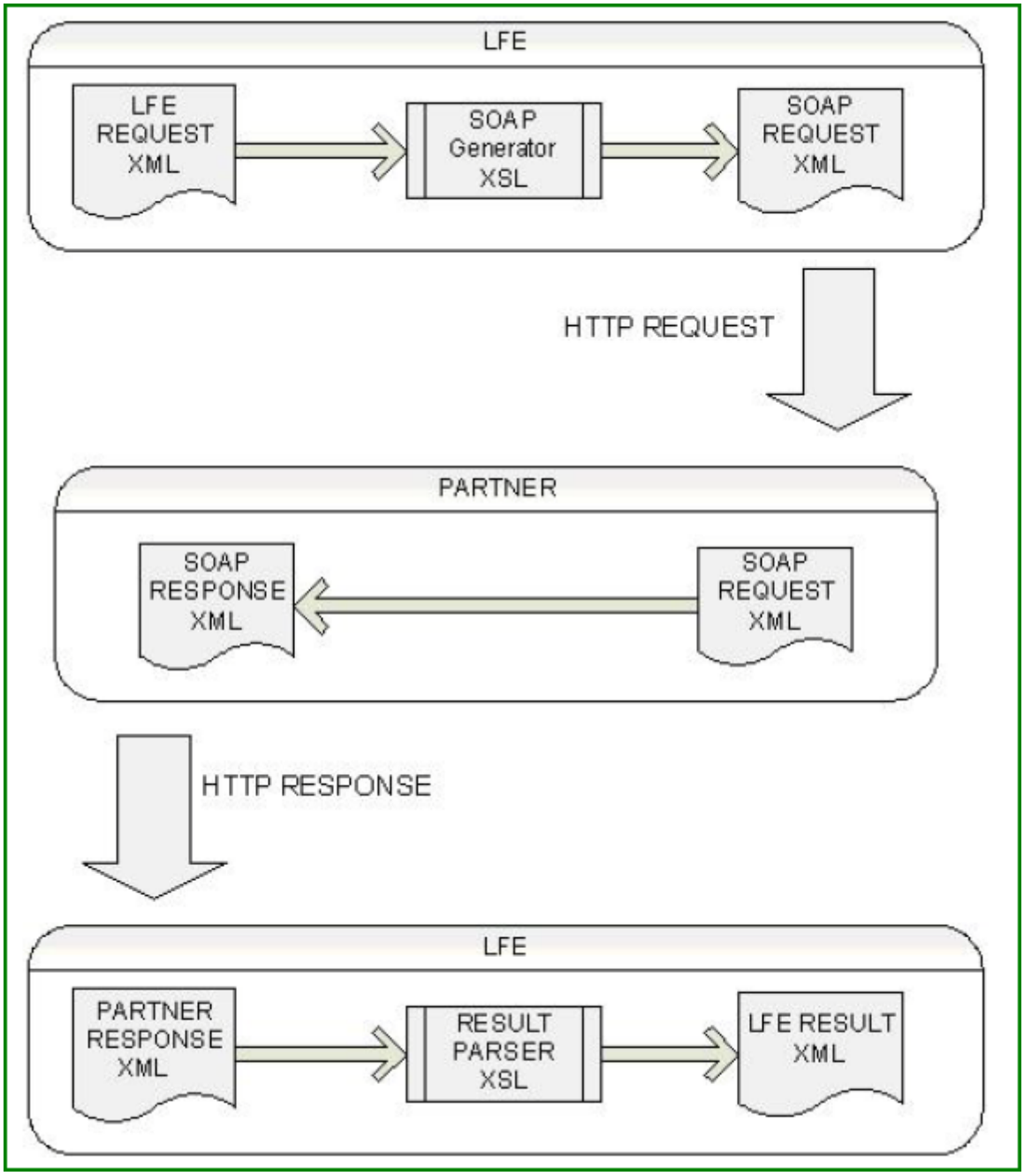
Methodology

The ability to communicate with a partner site using XML and SOAP protocol is what is commonly termed "Web Application Integration". The system sends data to the partner system in a specific XML format and uses SOAP to transmit the data. However the XML data that generates is in a format that needs to be understood by each partner. To keep it generic, XML requests are converted to the partner format using the SOAP generator XSL. The new XML format is then wrapped in a SOAP envelope to transmit to the partner over HTTP.

Similarly, the response is received from the partner is transmitted over HTTP using SOAP. The response XML is then converted to format using a Result Parser XSL.

Field Name	Description
SOAP Generator XSL	XSL to convert the XML format to the SOAP request that will be understood by the partner.
Results Parser XSL	XSL to convert the partner SOAP response XML to XML that is understandable.

Process Flow



RELATED TOPICS

SINGLE SIGN-ON (SSO)

OVERVIEW

BigMachines allows customers to utilize their corporate infrastructure for authentication and to automatically log into BigMachines without the need for multiple logins and re-authentication. There are two methods that can be used: Federated Authentication via SAML (Security Assertion Mark-Up Language) and Remote WebService. With Federated Authentication, users are managed in an Identity Provider or Portal (IDP) which supports SAML, so instead of visiting BigMachines directly, users access this Id Provider site before clicking on a link to access BigMachines. When using a Remote WebService, users are also managed with an IDP, but one that doesn't support SAML. Partner applications can submit a login request through a SOAP API call to the BigMachines login web service.

What is SAML 2.0?

SAML is an XML-based solution that provides a secure solution for exchanging authentication and authorization of user security information between IPD's and the applications like BigMachines.

What is an Identity Provider (IDP)?

Identity providers are sites or services that provide a security credential (such as an authentication or authorization assertion) on behalf of a user. In some cases this security credential may contain a set of attributes like a user's name or an employee number identifier.

What is an assertion?

An assertion carries authentication and authorization statements or credentials about a user that the IDP claims to be true. In this case, this information is sent to BigMachines to be verified against your SSO configuration.

For example, an assertion encodes the following information:

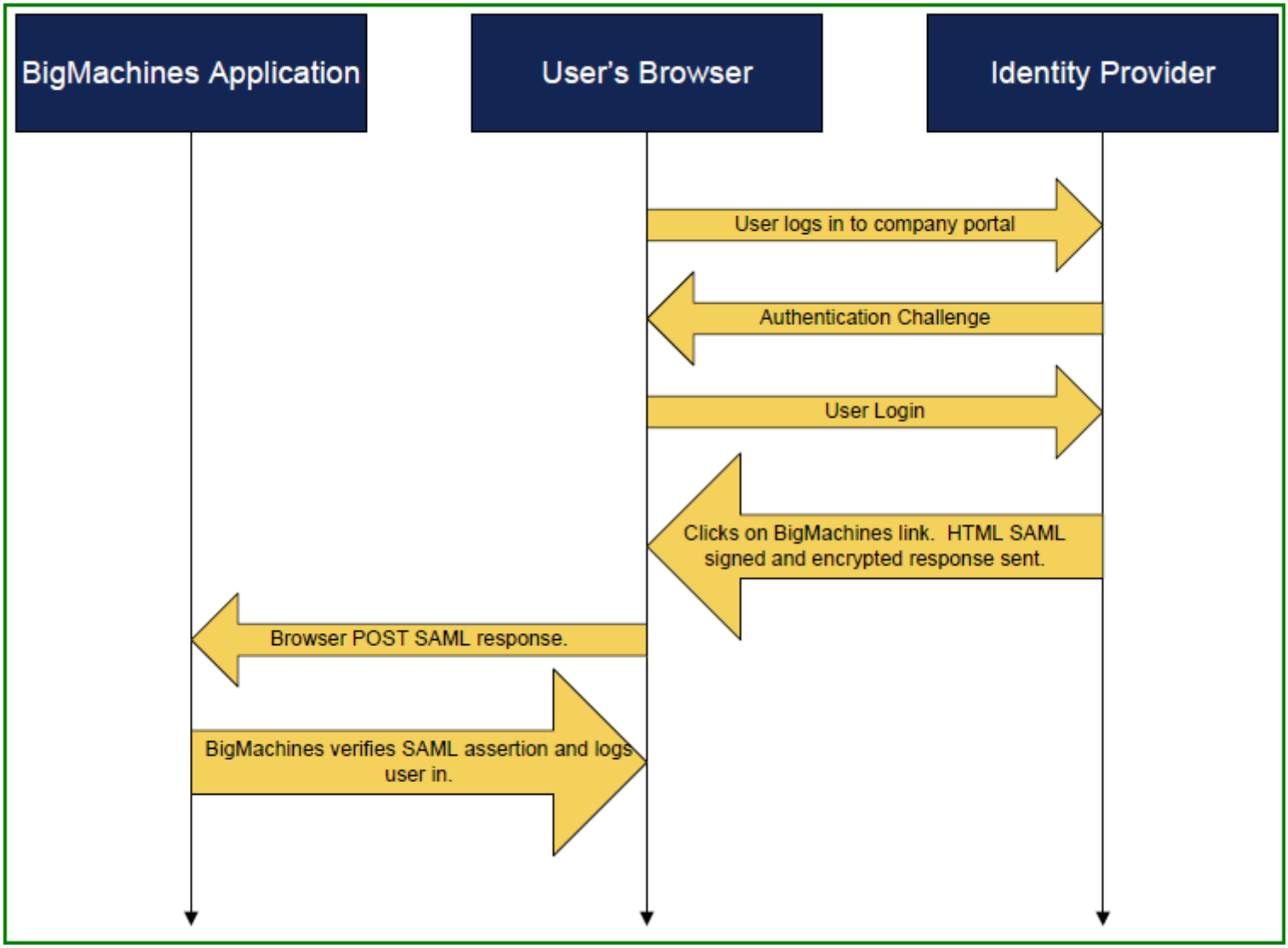
The assertion ("1d2v5") was issued at time "2004-12-05T09:22:05Z" by identity provider (https://idp.example.org/SAML2) regarding subject (user 123) exclusively for service provider (https://customer.bigmachines.com/SAML2).

The authentication statement, in particular, asserts the following:

The user identified in the <saml:Subject> element was authenticated at time "2004-12-05T09:22:00Z" by means of a password sent over a protected channel.

Likewise the attribute statement asserts that:

The user identified in the <saml:Subject> element is a staff member at this institution.



SSO Integration Criteria

Criteria	Value
Size	Small
Frequency	Triggered during each login
Format	SAML 2.0 or SOAP Web Service
Transmission Sychronization	Synchronous

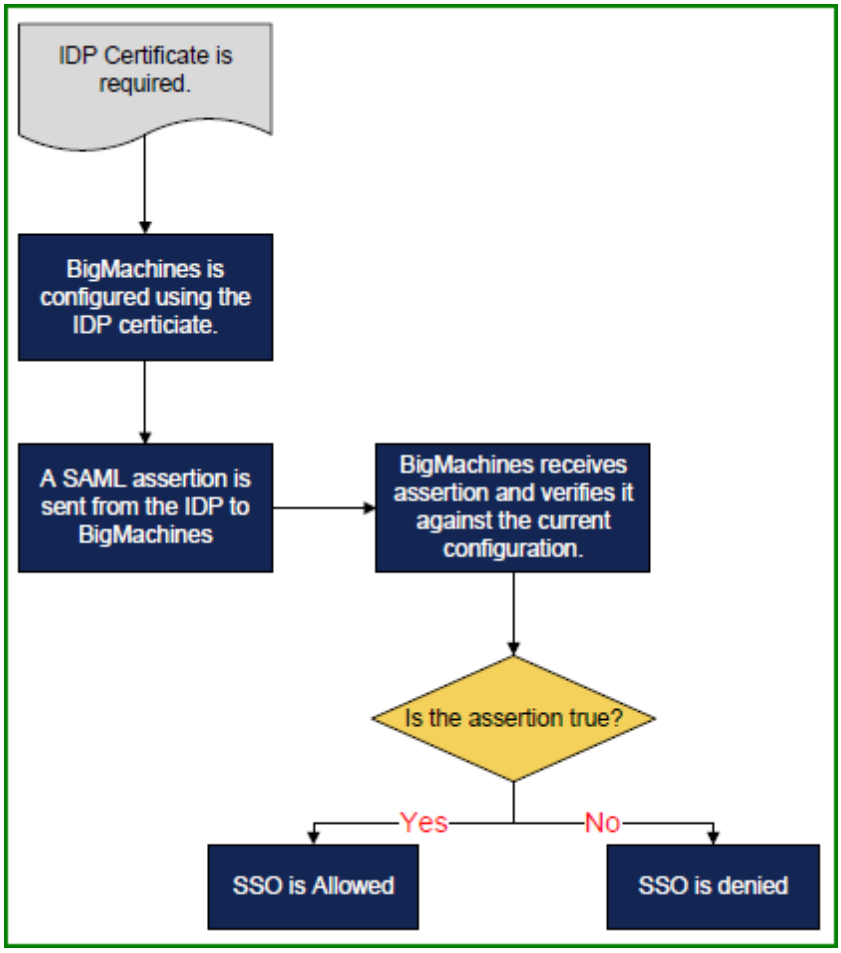
Sample Login XML

```
<?xml version = "1.0" encoding = "UTF-8"?>
<soapenv: Envelope
xlmns:soapenv=http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header><bm:category
xmlns:bm="urn:soap.bigmachines.com" soapenv:actor=http://schemas.xmlsoap.org/soap/actor/next soapenv:mustUnderstand="0">Security</bm:ca
xmlns:bm="urn:soap.bigmachines.com"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"soapenv:mustUnderstand="0">
<bm:schemaLocation>http://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/security/Security.xsd</bm:schemanLocation></bm:xsdInfo></soapenv
<bm:login
xmlns:bm="urn:soap.bigmachines.com">
<bm:userInfo>
<bm:username>MYUSERNAME</bm:username>
<bm:password>MYPASSWORD</bm:password>
<bm:sso>1</bm:sso>
</bm:userInfo>
</bm:login>
</soapenv:Body>
</soap:env:Envelope>
```

FEDERATED AUTHENTICATION VIA SAML

Before SSO via SAML can be used in BigMachines, work must be completed outside of BigMachines. The flowchart below illustrates what you need to obtain and how it relates to BigMachines.

Click here to view flowchart

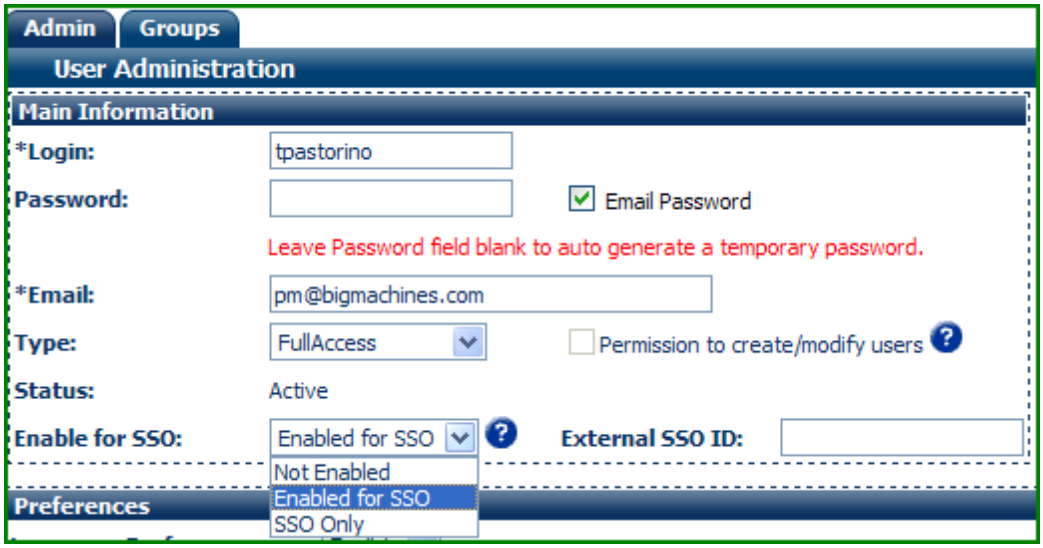


SAML Set-Up in BigMachines

1. Each user that needs to access the BigMachines application through SSO must have a user account. Make sure to complete required fields.

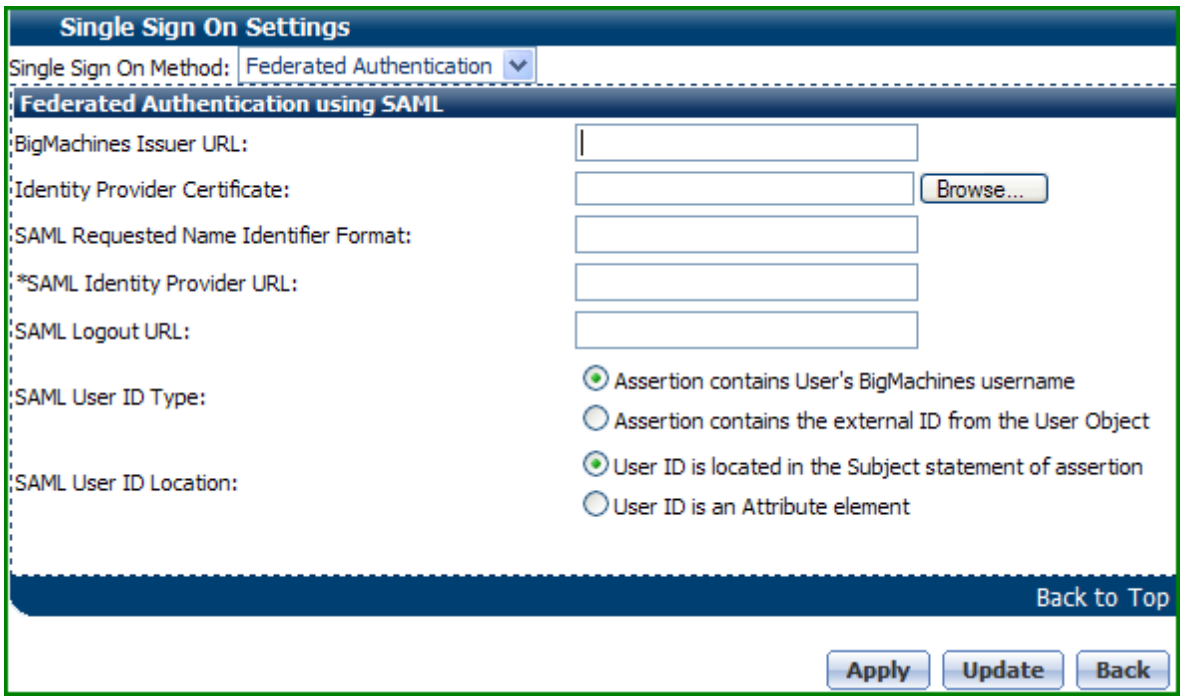
2. Select an option for 'Enable for SSO'. If the User is enabled for Single Sign-On login, then when the site is enabled for SSO, the user can access login via this method. If you know the user's External SSO ID, then that will be used for SSO, otherwise the User's Login will be used. There are three options to choose from:

    - **Not Enabled:** SSO is not available for this user, they must login to BigMachines directly.
    - **Enable for SSO:** SSO is available for this user AND they can login to BigMachines directly.
    - **SSO Only:** SSO is available for this user, however they cannot login to BigMachines directly.



3. Navigate to: Admin Home Page > Integration Platform > Single Sign On
4. From the Single Sign On Method drop-down menu, select Federated Authentication. This enables SAML SSO.
5. BigMachines Issuer URL: yourcompany.bigmachines.com
6. Click Browse to locate and upload an Identity Provider Certificate. This file details how to communicate with each particular Identity Provider. For example, information such as the IP login and logout will be found in this file.
7. If necessary, enter a SAML Requested Name Identifier Format into the text field. Each IDP supports different NameID formats. These can be found in the IDP configuration. This field is customizable to fullaccessusers, but if the field is left blank, we default to using the "transient" format.
8. Common NameID Formats:
    - urn:oasis:names:tc:SAML:2.0:nameid-format:persisten
    - urn:oasis:names:tc:SAML:2.0:nameid-format:transient
    - urn:oasis:names:tc:SAML:2.0:nameid-format:emailAddress
    - urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified
9. Enter the URL that identifies the SAML Identity Provider. This is a required step as all assertions that are sent to BigMachines need to have an Issuer value that is identical to this field.
10. Enter a SAML logout URL. When the user clicks on the logout link in BigMachines, they will be logged out and redirected to the URL in this field.
11. Select the SAML User ID type. This specifies what identifier an assertion contains when being sent to BigMachines: BigMachines username or the external ID referenced in Step 2.

12. Select the SAML User ID location. This specifies where in the assertion a user will be identified:
    - **Subject Statement:** User ID is located in the <Subject> statement of the assertion
    - **Attribute Element**: User ID is located in an <AttributeValue>, located in the <Attribute> of the assertion.
        - If this option is selected, a field named "Attribute Name" will become visible. Enter the value of the <AttributeName> that contains their User ID.



13. Click Apply to save your changes. Click Update to save your changes and return to the Admin Home Page. Clicking Back will return you to the Admin Home Page without saving your changes.



Your can see what your SAML looks like by going to: yourcompanyname.bigmachines.com/sso/spmetadata.jsp

---

REMOTE WEB SERVICE

Remote WebService allows user logged into partner applications to access BigMachines without having to login or re-authenticate. This method does not require that the IDP support SAML. In order to access BigMachines, the portal requires a customized SOAP call to the BigMachines login WebService.

SSO Sample Login XML

```
<?xml version = "1.0" encoding = "UTF-8"?><soapenv: Envelope
```

```
xlmns:soapenv=http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header><bm:category
xmlns:bm="urn:soap.bigmachines.com" soapenv:actor=http://schemas.xmlsoap.org/soap/actor/next soapen-
v:mustUnderstand="0">Security</bm:category><bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"soapenv:mustUnderstand="0">
<bm:schemaLocation>http://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/se-
curity/Security.xsd</bm:schemanLocation></bm:xsdInfo></soapenv:hdeader><soapenv:Body><bm:login
xmlns:bm="urn:soap.bigmachines.com">
<bm:userInfo>
<bm:username>MYUSERNAME</bm:username>
<bm:password>MYPASSWORD</bm:password>
<bm:sso>1</bm:sso>
</bm:userInfo>
</bm:login>
</soapenv:Body></soap:env:Envelope>
```

URL to redirect a user after login

https://XYZ.bigmachines.com/?sessionId=SESSION_ID&username=USER_LOGIN&sso=true

Verify:

- BigMachines is configured with the Remote WebServices SSO method.
- A BigMachines user is enabled for SSO in BigMachines and has an account in their IDP.
- Ensure that the login SOAP API contains an sso tag, for example: <bm:sso>1</bm:sso>

NOTES



For additional information, please log a ticket with Bigmachines Customer Support.

Users must exist in BigMachines, but SSO can be configured without a password if you are using Federated Authentication. When using Remote WebServices, BigMachines will still require a password.



An Exteranl ID is often used if a customer's security policy does not allow the username to be sent out of the IDP. Instead, the IDP sends out some other form of identification (employee number, alias, etc.) which has to match the External ID field in the User Object.



BigMachines SSO is configurable at User and Partner Org levels.

All URL parameters (sessionid, sso=true, usrname) are melded in order to login a user via Remote Web-services. Also, the SSO settings have to be set to Remote Webservices and the user must be enabled for SSO in order for the login to succeed.



BigMachines currently supports SAML 2.0, which is not backwards compatible.

RELATED TOPICS

DATA CUBE EXPORT

OVERVIEW

If you have a third party reporting or warehouse tool, you may want to export all of the data from BigMachines to this tool, so you can run reports. You can compare BM data with other data in their 3rd party system. In other words, Data Cube Export gives you a way to transmit BigMachines data, as a text file, to another location.

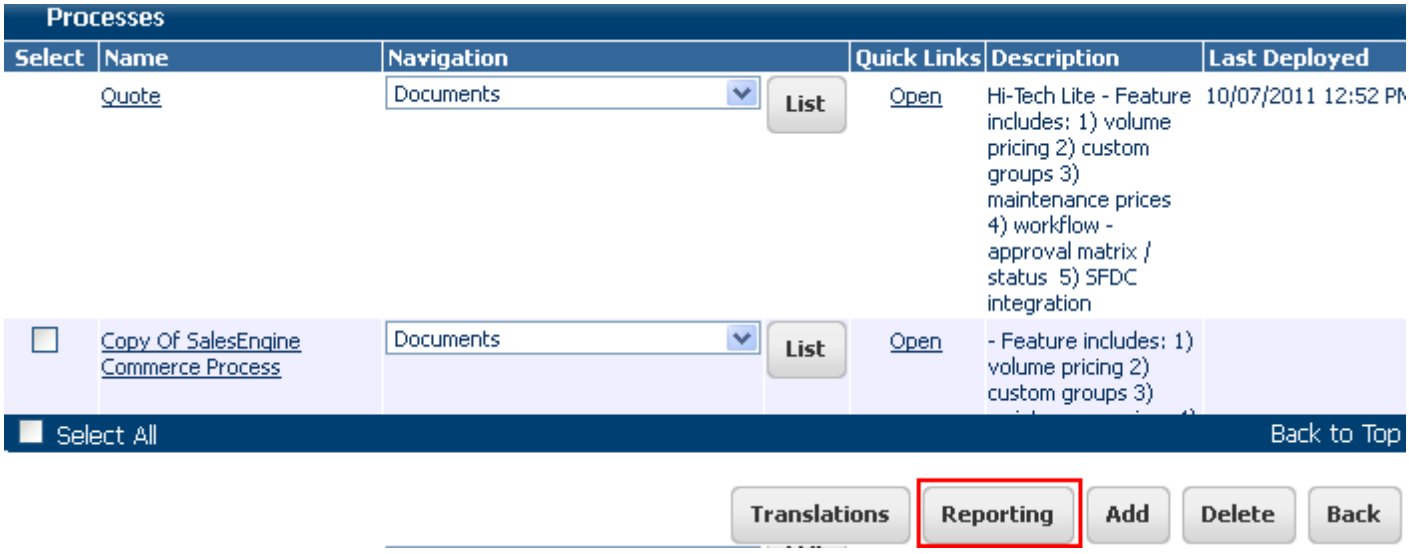
ADMINISTRATION

Setting up Data Cube Export

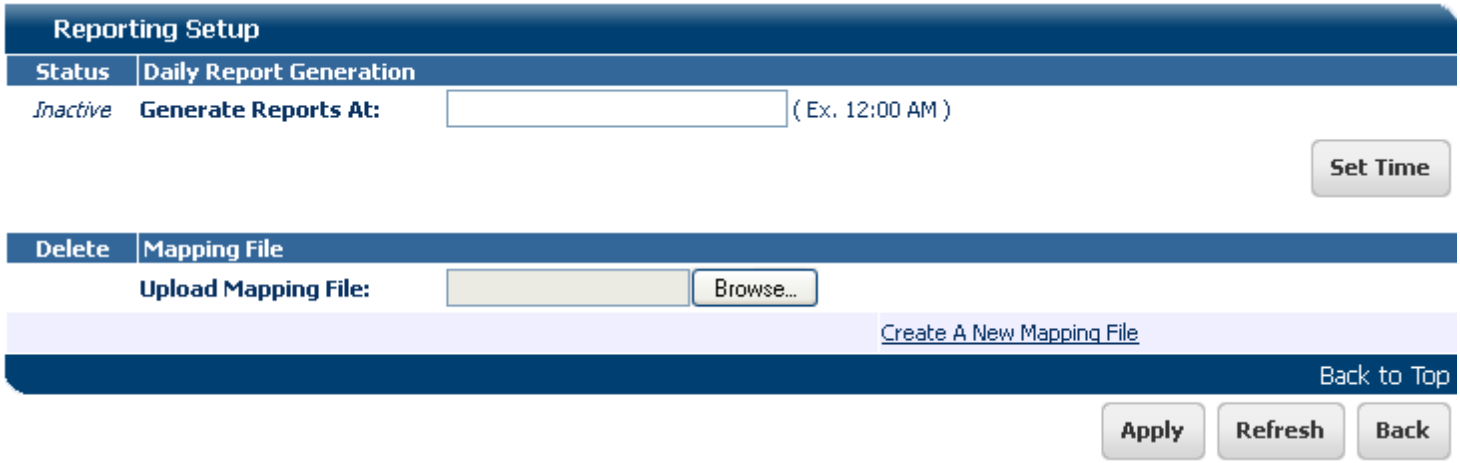
1. Contact **BigMachines OPS** to enable this functionality.

   Once enabled, you'll see a Reporting button appear in the Deployment Center for the commerce process you are reporting from. OPS will need to know where (which 3rd party system) to send the files (usually transmitted via SFTP). You'll have to work out encryption and decryption keys.

2. Click the **Reporting** button at the bottom of the Processes list page.



3. Click the **Create a New Mapping File** link to download the mapping XML.



4. The XML will look something like this:

```
-<mapping>
-<process var_name="TonisPizza_process">
-<document var_name="tonispizza_line">
<attribute var_name="_model_segment_id" db_name="bm_rt_4180720" />
<attribute var_name="_model_segment_name" db_name="bm_4180955" />
<attribute var_name="_model_product_line_id" db_name="bm_4180801 " />
<attribute var_name="_model_product_line_name" db_name="bm_4180751 " />
-</document>
-</process>
-</mapping>
```

5. Edit the **mapping file** in an XML editor. *This will tell you which data is included and what files it's placed in to, as well as the variable name of the columns.*

6. Update the **variable names for the database ID fields**. Once this is done, it should run seamlessly, should there be no additional changes. See the below XML for an example of the relationship between Commerce Attributes and the 3rd party database field IDs.

7. **Save and Upload** the completed mapping file back to BigMachines. To do so, click Browse next to the Upload Mapping File button.

8. Example XML (to Upload):

   The variable db_name represents the customer field and will have to be populated for each corresponding BigMachines attribute. *Please keep in mind that this is just a sample snippet of XML and does not represent what might be included in your mapping file.*

```
-<mapping>
-<process var_name="TonisPizza_process">
```

```
-<document var_name="tonispizza_line">
<attribute var_name="_model_segment_id" db_name="SEGMENT_ID" />
<attribute var_name="_model_segment_name" db_name="SEGMENT_NAME" />
<attribute var_name="_model_product_line_id" db_name="PRODUCT_LINE_ID" />
<attribute var_name="_model_product_line_name" db_name="PRODUCT_LINE_NAME" />
<attribute var_name="_model_id" db_name="MODEL_ID" />
<attribute var_name="_model_name" db_name="MODEL_NAME" />
<attribute var_name="tonispizza_line_size" db_name="MODEL_LINE_SIZE" />
<!--attribute var_name="config_attributes" db_name="CONFIG_ATTRIBUTES" / -->
<attribute var_name="_sequence_number" db_name="SEQUENCE_NAME" />
<attribute var_name="tonispizza_line_group" db_name="GROUP_LINE" />
<attribute var_name="tonispizza_line_price_1" db_name="PRICE_1" />
<attribute var_name="tonispizza_line_price_2 " db_name="PRICE_2" />
<attribute var_name="tonispizza_line_price_3 " db_name="PRICE_3" />
<attribute var_name="tonispizza_line_comments_1 " db_name="COMMENTS_1" />
<attribute var_name="tonispizza_line_comments_2 " db_name="COMMENTS_2" />
<attribute var_name="tonispizza_line_comments_3 " db_name="COMMENTS_3" />
<attribute var_name="_price_quantity" db_name="QUANTITY" />
<attribute var_name="tonispizza_line_list_price" db_name="LIST_PRICE" />
<attribute var_name="tonispizza_line_base_price_discount" db_name="BASE_PRICE_DISCOUNT" />
<attribute var_name="tonispizza_line_adder_discount " db_name="ADDER_DISCOUNT" />
<attribute var_name="tonispizza_line_multiplier " db_name="MULTIPLIER" />
<attribute var_name="tonispizza_line_net_price" db_name="NET_PRICE" />
<attribute var_name="tonispizza_line_profit_margin " db_name="PROFIT_MARGIN" />
<attribute var_name="tonispizza_line_sell_price" db_name="SELL_PRICE" />
-<document var_name="tonispizza_quote">
<attribute var_name="_customer_id" db_name="DISTRIBUTOR_ID" />
<attribute var_name="_bill_to_first_name" db_name="DISTRIBUTOR_FIRST_NAME" />
<attribute var_name="_bill_to_last_name" db_name="DISTRIBUTOR_LAST_NAME" />
<attribute var_name="_bill_to_company_name" db_name="DISTRIBUTOR_COMPANY_NAME" />
<attribute var_name="_bill_to_address_1" db_name="DISTRIBUTOR_ADDRESS_1 />
<attribute var_name="_bill_to_address_2" db_name="DISTRIBUTOR_ADDRESS_2" />
<attribute var_name="_bill_to_city" db_name="DISTRIBUTOR_CITY" />
-</document>
-</process>
-</mapping>
```



BigMachines 12.2 has a new field for data cube export that defines the number of days history that needs to be downloaded (number_of_days_of_data_to_export). For example, if the field is populated with a '4', then the data cube export would contain the last four days. Please contact OPS to enable this setting.

Data Cube Delta Days

This field defines the number of days history that needs to be downloaded. For example, if the field is populated with a '4', then the data cube export would contain the last four days of data.

NOTES



If you wish to utilize the Data Cube Export Delta Days feature, create a case to enable or modify this setting.



This is an add-on service. Please contact your Project Manager to discuss.

RELATED TOPICS

STANDARD XML OUTPUT

OVERVIEW

Here is an overview of BigMachines standard XML output.

```
<quote_process process_var_name="quote_process_bmClone_16"document_name="Quote" document_var_name="quote_process" document_number="1" data_type="0" bs_id="4607094" currency_pref="USD" buyer_company_name="Training" supplier_company_name="Training" buyer_user_name="superuser">

<quote level commerce attributes>Listed as variable names in the same order as the Attribute List page.</quote level commerce attributes>

<opportunityName_quote>Test Printing</opportunityName_quote>

<sub_documents>

 <line process process_var_name="quote_process_bmClone_16" document_name="Line" document_var_name="line_process" document_number="2" data_type="3" bs_id="4607094" currency_pref="USD" buyer_company_name="Training" supplier_company_name="Training" buyer_user_name="superuser">Here is where you will find line-level commerce attributes

</line process>

</sub_documents>

<current_user>tpastorino</current_user>

</quote_process>
```

ADMINISTRATION

Color Code

Color	What does it represent?
Purple	These are the XML attribute names. Please note that some commerce attribute names are also shown in purple, as these are the same as the XML attribute names.
Blue	This represents the variable data and is subject to change from quote to quote.
Black	This represents the variable names of the information found in the xml tags <quote_process> and <line_process>.
Gray	This represents data found within the quote and line-level commerce attributes. Shown above in the tag <opportunityName_quote>, you'll see the name of the opportunity is 'Test Printing'.

Variable Tags

Many of the variable names found within the <quote_process> and <line_process> tags are similar. For example, process_var_name refers to the variable name of the entire commerce process.

Variable Name	What does it represent?
process_var_name	The variable name for the commerce process.
document_name	The type of document it is: Quote or Line.
document_var_name	The variable name of the document. For example, 'line_process'.
document_number	The quote level document is 1 and each subsequent line item is given a document number in order.

Variable Name	What does it represent?
data_type	0 - Represents the enter quote 2- Represents a model-level line item 3 - Represents a recommended item or an item from a parts search
bs_id	The transaction ID.
currency_pref	The preferred currency of the host company.
buyer_company_name	The company name of the buyer.
supplier_company_name	The company name of the supplier.
buyer_user_name	The user name of the buyer.

NOTES



Please note that there is not an XSL stylesheet associated with this, so the structure might vary slightly depending on how your application is configured.

RELATED TOPICS

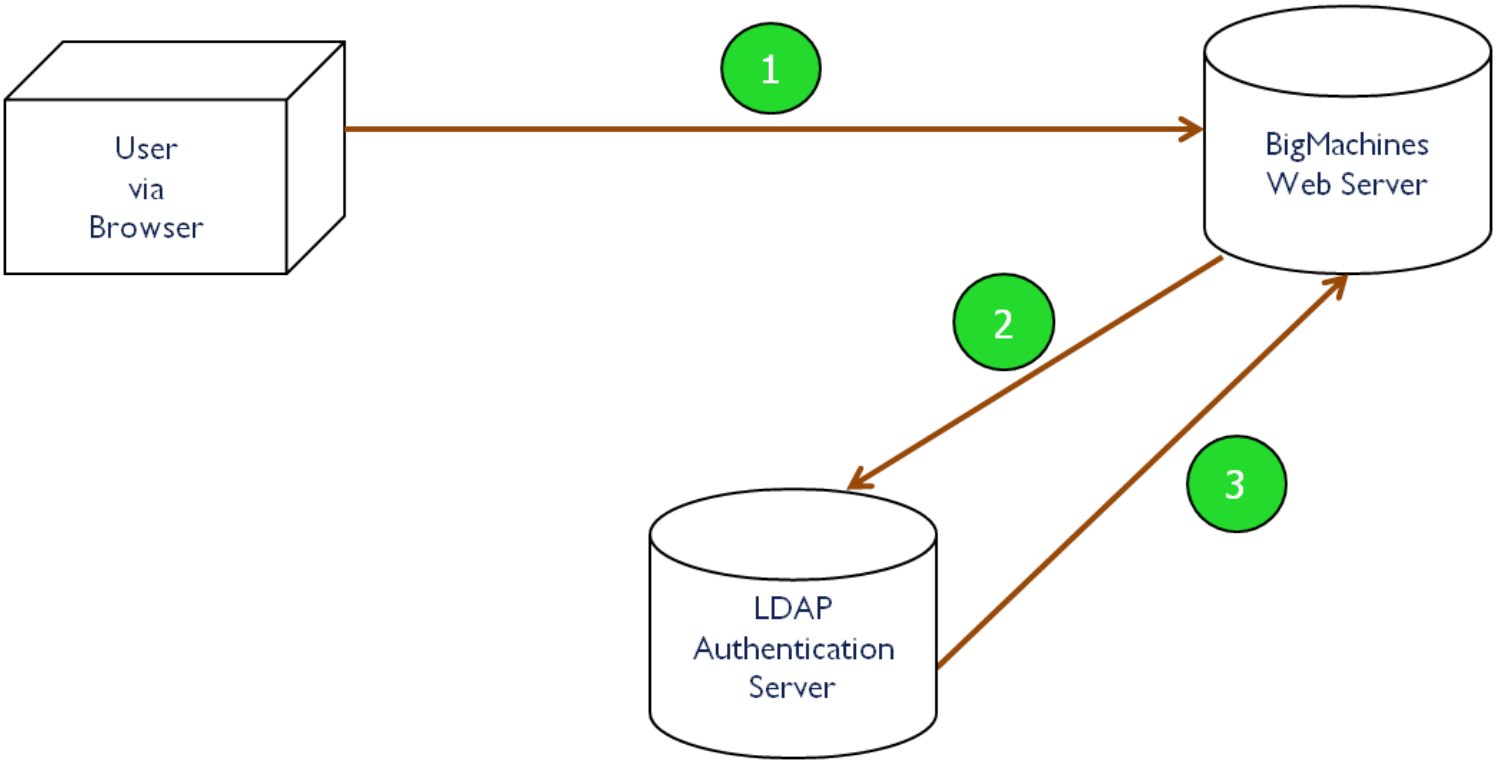
SETTING UP LDAP

OVERVIEW

LDAP (Lightweight Directory Access Protocol) is used to look up encryption certificates, pointers to printers and other services on a network, and provide "single sign-on" where one password for a user is shared between many services. LDAP is appropriate for any kind of directory-like information where fast look-ups and less frequent updates are the norm. LDAP also defines permissions set by the fullaccessusers to allow only certain people to access the LDAP database and optionally, keep certain data private. *BigMachines supports integration with any supplier's LDAP for authentication.*

How does it work?

1. The user logs in.
2. BigMachines sends authenticated query to LDAP.
3. LDAP confirms the authentication from BigMachines and logs the user in.



ADMINISTRATION

Setting Up LDAP

1. Synch required information.



The connectivity information will be used to connect to the supplier's LDAP. The search criterion will help in retrieving the appropriate limited set of user,s as there can potentially be many users in a company's LDAP. If a given LDAP field has its priority marked, then the LDAP value for that field takes priority over value in the BMI user record.

What?	Crieria	Description
Connectivity Information for the Supplier's LDAP	ldap_initial_ctx_factory	
	ldap_provider_url	URL of the LDAP server. Should follow the format of "ldap://DNS of LDAP:Port Number" OR "ldap://IP Address:Port Number". Either the DNS or IP Address can be used, but the Port Number must be included.
	ldap_binddn_username	Username for the LDAP server. User must have access to the entire directory that will be searched.
	ldap_binddn_password	Password for the above LDAP server.
OU	Organizational Unit	
Search Criteria in LDAP Tree	ldap_search_scope	Details of the depth of the search in the LDAP tree structure for Organizational Unit. Can be either 0 (base-level), 1 (first level children under the base) or 2 (Entire subtree underneath the base).
	ldap_leaf_nodename	Filter criteria. For Microsoft AD, this must be objectClass. For other servers, this can list the filter criteria.
Additional Information	ldap_passwrd_encyrption	Encryption type for passwords being sent for authentication. Common is MD5.
	ldap_login_synch	Setting to turn on LDAP-BMI data synch for every time a user logs in. true = BMI data is updated with LDAP info every time a user logs in. false = No sync or login. Sync must be manual, within BMI. Recommended by BigMachines.
Connectivity information provided by BigMachines	LDAP field mapping to BMI database column	
	LDAP field value format mapping to BMI database compatible value	
	LDAP field priority marker over BMI field, for updating the BMI DB record	
	LDAP field priority marker over BMI field, for updating the BMI DB record	
	Default value for each field in BMI database	

2. Sample LDAP Properties Text

```
########### LDAP Properties ############
ldap_initial_ctx_factory=com.sun.jndi.ldap.LdapCtxFactory
ldap_provider_url=ldap://xyz.pdq.btt.com:636
ldap_binddn_username=cn=t3_Xabnie,ou=Application,dc=cat,dc=com
ldap_binddn_password=t3!Xabnie
ldap_search_scope=2
ldap_leaf_nodename=
ldap_passwd_encryption=
ldap_login_synch=true
#######################################
```

LDAP Information

The synch-up information between LDAP and BigMachines is available in the form of an XML Document.

Pre-conditions for LDAP user login

1. The following conditions are to be met before the LDAP user trying to login to the system, gets successfully authenticated against the supplier's repository –
2. The BMI Admin has to first upload the LDAP schema XML providing the correct LDAP OU information and user attributes mapping to lookup (search) for LDAP users in the BMI application for mapping.
3. The LDAP user should have an email in the supplier's repository when the ldap_priority is set to true.
4. The fullaccessuser should have successfully mapped the LDAP user to BMI user
5. The LDAP user will then login with the BMI login name and his LDAP password.

LDAP User Search Page

The fullaccessuser will search on a list of LDAP users based on the LDAP profile attributes under the search criteria (list of OUs) available in the configuration. OUs refer to the list of organizational units in LDAP. The admin can multi-select the OUs. This search will retrieve a list of qualified users. This will forward the admin to the next page where he can view the existing mappings for users and add/modify them.

Map LDAP user to BMI user - available options

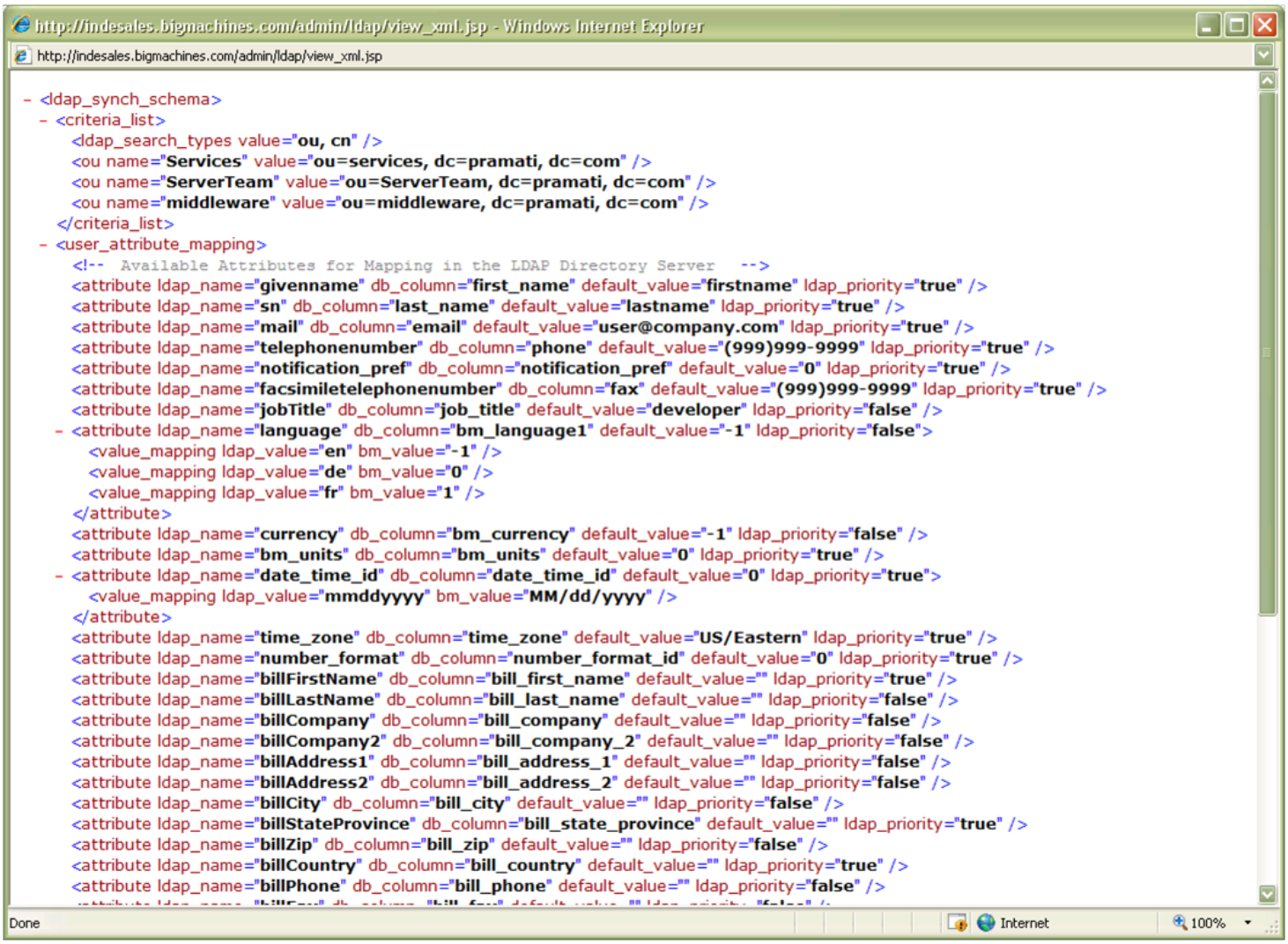
- Map LDAP users to existing BMI users
- Erase existing mappings between LDAP and BMI users
- Synch up individual or a bunch of mapped users with their LDAP profile info

Any LDAP user will be mapped to one and only one BMI user. The admin can synch all the mapped users or selected mapped users. Selecting them and clicking on "unMap" can unmap the mapped users. The mapping status is also shown via the Map/Un-map button. The LDAP to existing BMI user mapping can be done by selecting the BMI user from the pull-down list. The view the

list of companies and the list of related users to choose from. The Synchronize buttons does the synch-up of the profile-related info.

LDAP Schema Manipulation

The fullaccessuser will be able to view or update the LDAP schema. The “Display Sample LDAP Schema” and “Display LDAP Schema” options as shown below on the search inputs page. The admin will also be able to upload a new schema from here. Once the new schema is uploaded, the impact is immediate, i.e., any subsequent synch operations will pick up the new schema. The sample schema that is currently in effect can be viewed using the display link besides the upload button.



LDAP Maintenance and Troubleshooting

Editing the Synch-Up Schema

1. Additional OUs can be added to the criteria list.
2. Name = the label name for the OU. This is displayed in the BMI UI and used for searching.
3. Value = the LDAP OU value. Should contain a ou and dc information.
4. No additional attributes can be added for search criteria.
5. No BMI db_column_name settings can be changed.
6. The ldap_name value can be updated for attributes. This will change the value to select in the BMI UI as well as change the synching that occurs between BMI and the LDAP server.

LDAP Mapping Settings and User List Page

The LDAP mapping status for a BMI user is displayed on the user list page as shown below. It can be “Mapped” or “Deleted in LDAP”. The admin has to explicitly update the LDAP status to get the latest info on any given mapping. The fullaccessuser can performing the following tasks:

1. Update LDAP Status will check on the LDAP to see if the user still exists (may have been deleted.
2. For a user that’s deleted in LDAP, the fullaccessuser can choose to delete the mapping in the system by selecting the user and pressing UnMap

---

Synch-Up Single User on Users List Page

Synch-up is available only for an existing LDAP user in BMI database.

When the admin does a synch-up on a given user, only the non-priority marked fields will be taken from the UI input. The rest is picked up from LDAP profile information. This implies that the priority marked fields should be grayed out on the user details page. The synch button will be shown only for an LDAP user.

RELATED TOPICS

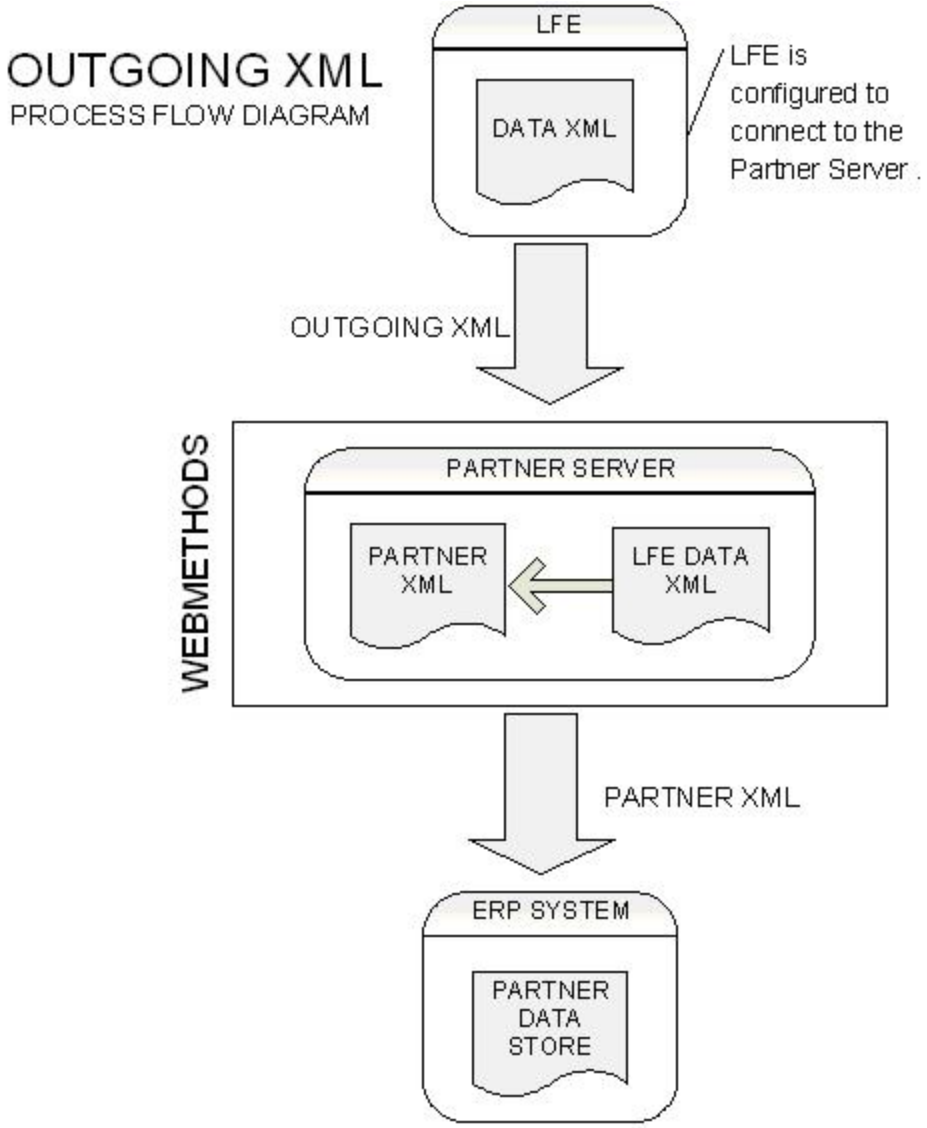
INTEGRATING WITH AN ERP (BACKEND)

OVERVIEW

Backend integration (for example, and an ERP system) is most often achieved using a third-party integration tool called WebMethods. The bulk of backend integration is accomplished by exchanging XML data between and an ERP system. **There are 2 types of back-end integration:**

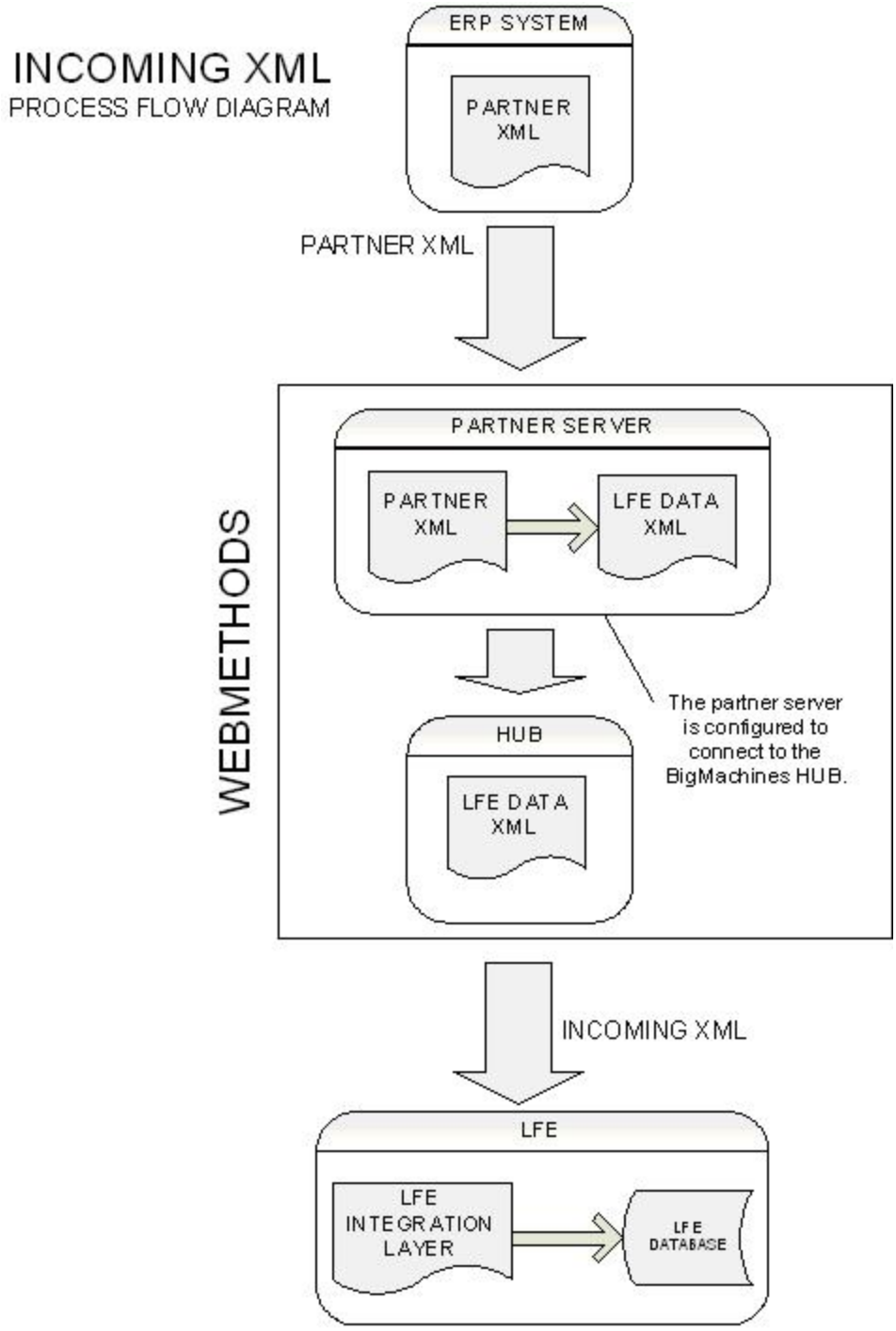
Outgoing XML

This refers to all XML sent from to the partner server (Client WebMethods box) and from the partner server it is sent to the ERP system. Note: The partner server can be configured to modify the XML into a format that is understood by the ERP system.



Incoming XML

This refers to all XML sent from an ERP system to the hub (WebMethods box) and from the hub it is forwarded to the application. Note: The hub can be configured to modify the partner XML into a format that is understood by. You can use either the outgoing XML method or incoming XML method or both.



ADMINISTRATION

Best Practices

- Use built-in services instead of JavaScript when possible.
- Use string templates instead of concatenation.
- Insert database records all at once, not individually.
- Close database connections after insertion is completed.

NOTES



Web Methods is the preferred enterprise application integration tool.

RELATED TOPICS

FTP AUTOMATION

OVERVIEW

FTP Automation allows uploads (similar to Bulk Uploads functionality) to an FTP server. Scheduled uploads can be automatically imported into a BigMachines site. *Automated uploads can include:*

- Parts
- Part Extended Descriptions
- Data Tables
- File Manager Files

FTP Automation is useful for BigMachines sites that require stored data to be refreshed regularly (e.g. updates to pricing tables from an ERP system), especially if BigMachines is the Parts Master (for SFDC Parts Integration) and the Parts data is pushed from an ERP system to the BigMachines site.

ADMINISTRATION

Setting up FTP Automation

1. Consult OPS for the creation an FTP profile and its corresponding "automated" folder on the FTP server.
2. Prepare the files that will be uploaded.
    1. Determine the format of the CSV files that will be uploaded. *Usually, the CSV format can be obtained by down-loading the relevant CSV file from Bulk Data Services.*
        - **Data Tables:** The Data Table CSV format used with FTP Automation differs from the file format used when Data Tables are uploaded through the UI. *The Data Table CSV file should not include any header rows when using FTP Automation.*
    2. After exporting a Data Table via the UI, delete the first 5 rows (`_start meta data`, column headings, data types, descriptions, and `_end meta data`) before uploading the table using FTP Automation.
    3. Place each file that will be uploaded into a separate .zip file.

---

3. Prepare the Upload List (upload_list.xml).

The `upload_list.xml` file lists the uploaded files in the order they will be processed. **Example Upload List file:**

```
<?xml version="1.0" encoding="UTF-8"?>
<files>
<file>
<name>Filename.zip</name>
<type>REGULAR</type>
</file>
```

```
</files>
```



For Data Table CSV uploads, the type is DATA_TABLES_CSV. For all other uploads, the type is REGULAR.

4. Upload the files to the automated folder on the FTP server.
    1. Upload the .zip files followed by the `upload_list.xml` file.
    2. Test the uploads manually.
    3. Test the automated process.



Testing the manual & automated process ensures that the schedule allows sufficient time to process one upload before the next upload is sent.



If an `upload_list.xml` file lists ZIP files that have not been uploaded, any ZIP files that were uploaded will still be processed correctly.



Files are not deleted from the temporary folder if the scheduled upload time passes and no `upload_list.xml` file is uploaded. *However, if a new `upload_list.xml` file is uploaded (e.g. for a new set of files in the next scheduled upload), it will overwrite the previous `upload_list.xml file`, and the corresponding ZIP files will not be uploaded to BigMachines.*

Additional Data Table functionality

When uploading data tables, records in the upload are appended to the existing table in BigMachines.

Data Table commands

In addition to uploading Data Tables, actions such as Delete and Deploy can be triggered using the automated upload.

Action	Files to Upload
Delete	• upload_list.xml (this will specify Delete.zip with type "REGULAR") • Delete.zip (this will contain a delete.xml file)
Upload	• upload_list.xml (this will specify Table.zip with type "DATA_TABLES_CSV") • Table.zip (this will contain a table.csv file)
Deploy	• upload_list.xml (this will specify Delete.zip with type "REGULAR") • Delete.zip (this will contain a delete.xml file)



These processes must run in the correct sequence, either by uploading the Delete, Upload, and Deploy files during consecutive scheduled upload events, or by loading at the same time listed in the correct order in the `upload_list.xml` file.

Example Delete XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<record_list>
<site_version><![CDATA[12.2.0]]></site_version>
<each_record>
<category>bm_script_data</category>
<action>delete_all</action>
<table_name>pricing</table_name>
</each_record>
</record_list>
```

Example Deploy XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<record_list>
<site_version><![CDATA[12.2.0]]></site_version>
<each_record>
<category>bm_script_data</category>
<action>deploy</action>
<table_name>pricing</table_name>
</each_record>
</record_list>
```



For more information on Data Tables, see Data Table Overview.

NOTES



Consult OPS before setting up FTP Automation.

RELATED TOPICS

ACCOUNTS

OVERVIEW

Accounts is a tool for supporting collaborative customer management in the BigMachines system. Customer records can be individually created in the Accounts database or they can be bulk uploaded into the system. Once customer records exist in Accounts, they are accessible to sales agents, channel agents and full access users.

All users that have access to customer accounts in Accounts also have permission to make updates or additions to Account records. Accounts records can contain general customer information, multiple address records for billing and shipping data, and comments. To collect more specific information, fullaccessusers can generate supplemental fields for capturing custom data. For more information, see Adding Custom Attributes.

A single Account record can be associated with one or more partner users. Associations are necessary if you plan to integrate Accounts with other areas of administration, such as configuration rules and commerce documents. To learn more about associations, see Defining Associations.

Accounts can also be integrated with commerce to auto-fill commerce document fields. Integrating Accounts with commerce speeds up document creation time by eliminating redundant manual data entry.

Managing User Accounts

The Accounts section maintains customer records. Using this page you can:

- Search for Customer Records
- Add Accounts Manually or through Integration
- Import Account Files
- Export Master Data and/or Address Information
- Create Custom Attributes for Accounts
- Customize Search: See Setup Account Search Criteria for more details.
- Customize Search Results: See Customizing Search Results for more details.

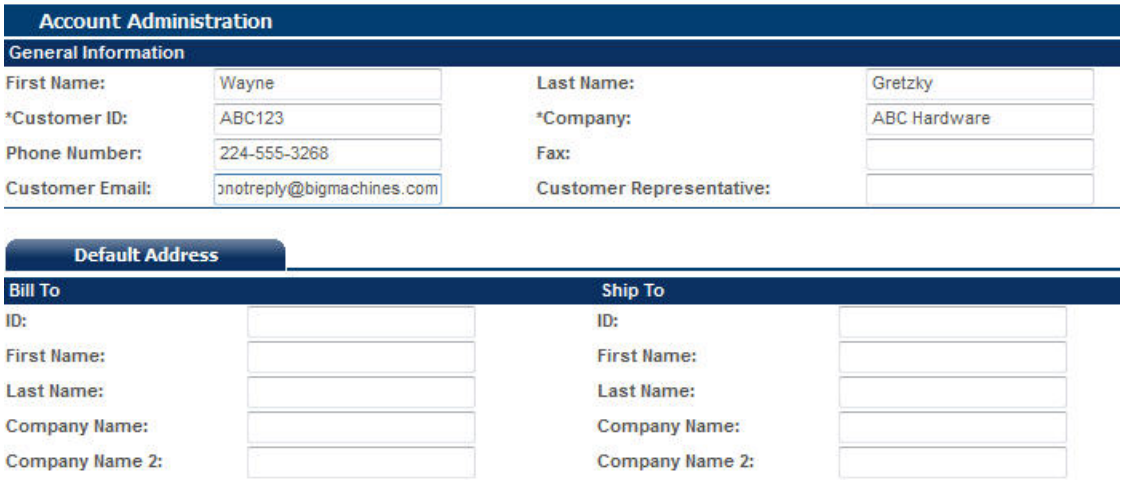
ADMINISTRATION

Creating an Account (Manually)

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Add** button.
3. Enter **General Account Information** and provide **Default Bill To/Ship To Addresses.**

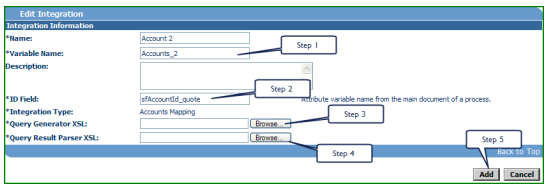
Customer ID and **Company Name** fields are required to create an Account.

4. In Additional Information, complete any custom attribute fields.
5. Click **Add**.



Creating an Account (Integration)

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click **Integration** to be taken to the Integration for CRM Page.
3. Click **Add**.
4. Enter a Name, Variable Name and Description for the new integration.
5. In the ID field, enter the attribute variable name from the main document of a process.
6. Click **Browse** to locate and add a Query Generator XSL file.
7. Click **Browse** to locate and add a Query Result Parser XSL file.
8. Click **Add** to add the new integration or click Cancel to go back to the page for managing integrations.

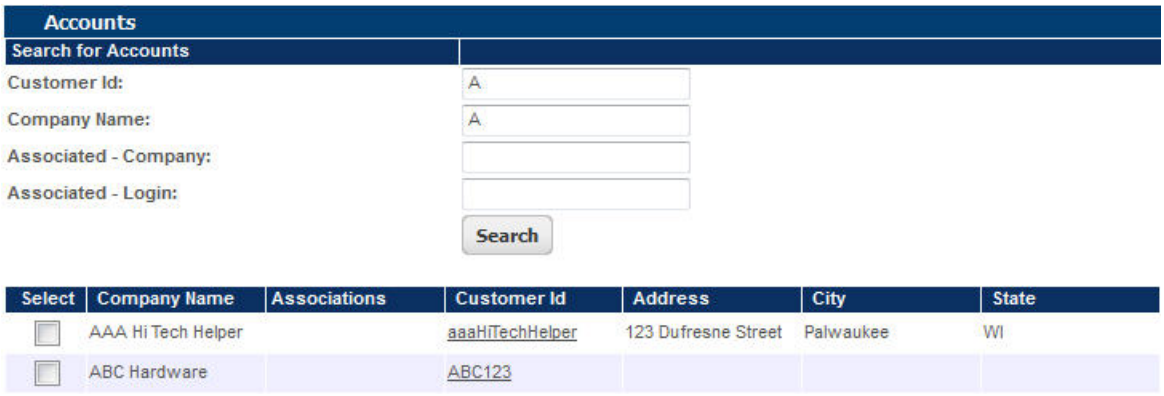


A Commerce Transaction ID field will appear after you've saved the Intgration.

9. Enter a commerce **Transaction ID** and click **Show Query Input XML** or Preview SOAP XML for additional integration details.

Managing Existing Accounts:

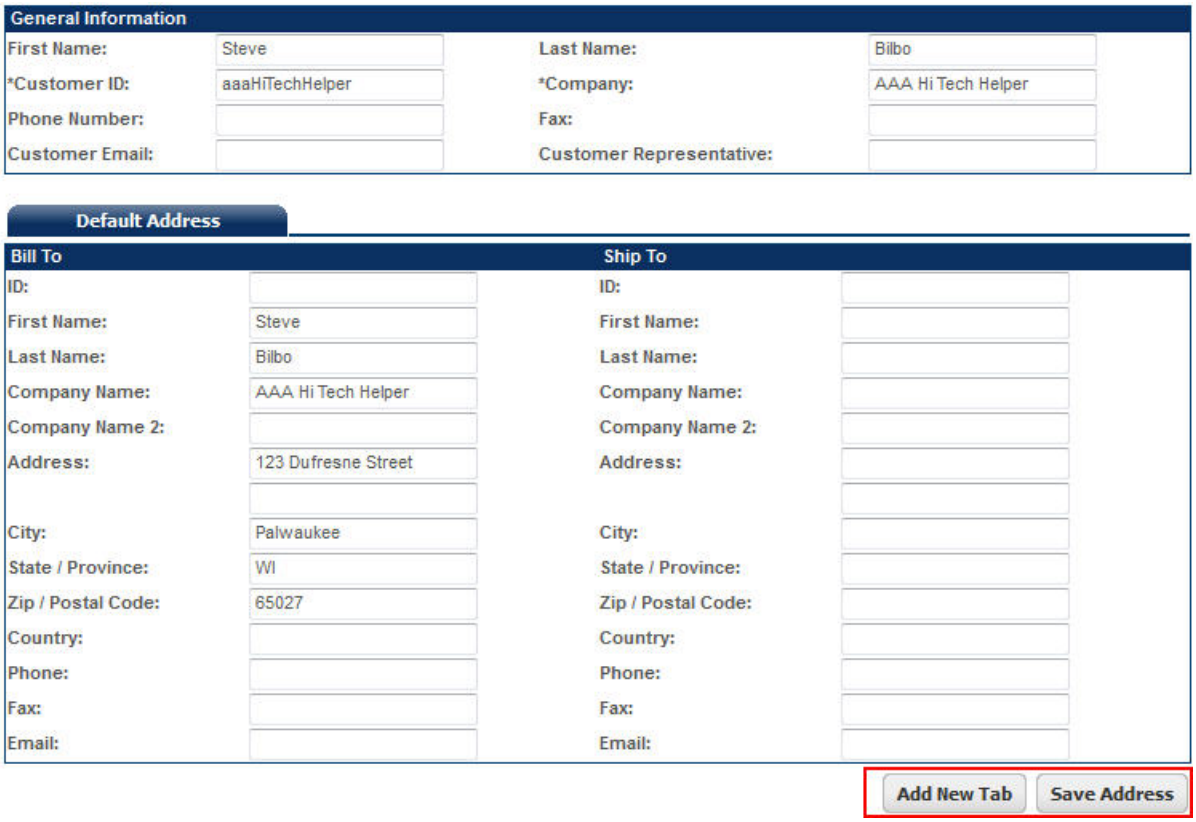
1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Search for the **Account** by Customer ID, Company Name, Associated-Company or Associated-Login.
3. Click **Search**.
4. Click on the **Customer ID**of the Account you want to edit.



5. Edit or enter account information in the fields provided.
6. Click **Add New Tab** to add an alternate address to new and existing Accounts records.
   - Enter a **tab label** for the new address, enter alternate address data, and click Save Address. Click on Delete Address when it is no longer required or valid.

     If you accidentally click the Add New Tab button instead of the Save Address button, cancel the operation. Otherwise, the new address record will be lost.

   - Alternate addresses appear as separate address tabs and can be viewed or modified by clicking the tab name.



7. Click on **Associations** to define account associations. See Defining Associations for more information.
8. Click **Update** to Save changes to information or click Back to return to the Accounts page without saving.

Importing Account Files

1. Prepare a .CSV file for Import.
2. Use the standard bulk upload actions to upload information.

> Click **Export Master Data** or **Export Address Data** for a template that you can use for the upload.

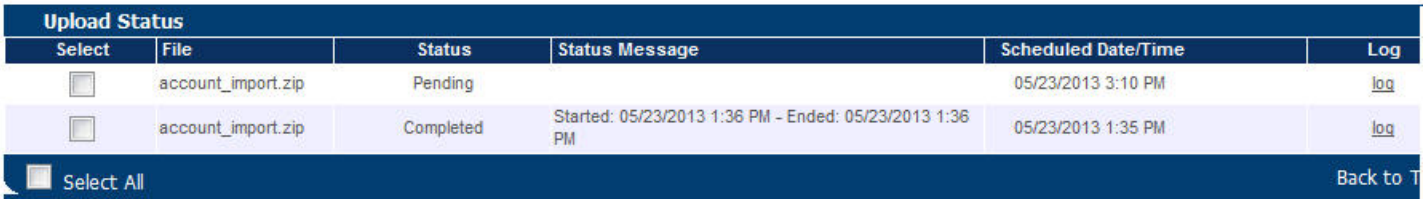
4. Enter information into the available fields. Customer ID (customer_id) and Company Name (company_name) are required.

	A	B	C	D	E	F	G	
1	Accounts Master							
2								
3	update	first_name	last_name	customer_id	company_name	customer_rep	phone	er
4	add	Steve	Bilbo	aaaHiTechHelper	AAA Hi Tech Helper			
5	add	Wayne	Gretzky	ABC123	ABC Hardware		224-555-3268	do
6	add	Ned	Finterwink	nedsComputerShed	Neds Computer Shed			
7	add	Beazus	Stankus	nerdherd	Nerd Herd			
8	add	Owen	Lawson	newOrleansDigital	New Orleans Digital			
9	add	Jason	OReilly	northeasternRack	Northeastern Rack			
10	add	Ron	Swanson	steelToolWerks	Steel Tool Werks			
11	add	Toni	Pastorino	ABC234	ABC Hardward - San Mateo		224-555-3269	do

5. Save the file and create a .zip file.
6. Navigate from Admin Home Page > Integration Platform > Accounts
7. Click the **Import** button.
8. Click the **Browse** button and choose a file.



9. Click **Add**. Repeat Steps 1-9 for all files to be imported.
10. Select the **file** and click the **Upload** button.
11. Click the **Refresh** button to see the status of the import.



12. Click the **log** to verify the upload has occurred without errors.

Export Master Data

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Export Master Data** button.

3. Select an option to Open or Save the file. The data will be downloaded as a .zip file.
4. Open the .zip file and double-click on the .csv file to open it.

	A	B	C	D	E	F	G	
1	Accounts Master							
2								
3	update	first_name	last_name	customer_id	company_name	customer_rep	phone	e
4	add	Steve	Bilbo	aaaHiTechHelper	AAA Hi Tech Helper			
5	add	Wayne	Gretzky	ABC123	ABC Hardware		224-555-3268	d
6	add	Ned	Finterwink	nedsComputerShed	Neds Computer Shed			
7	add	Beazus	Stankus	nerdherd	Nerd Herd			
8	add	Owen	Lawson	newOrleansDigital	New Orleans Digital			
9	add	Jason	OReilly	northeasternRack	Northeastern Rack			
10	add	Ron	Swanson	steelToolWerks	Steel Tool Werks			

Export Address Data

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the Export Address Data button.
3. Select an option to Open or Save the file. The data will be downloaded as a .zip file
4. Open the .zip file and double-click on the.csv file to open it.

	A	B	C	D	E	F	G	H	I	J	K	
1	Accounts A	ddresses										
2												
3	update	customer_	tab_label:	tab_label:	tab_label:	tab_label:	billing_firs	billing_last	billing_con	billing_con	billing_add	billi
4	add	aaaHiTech	Default Address				Steve	Bilbo	AAA Hi Tech Helper		123 Dufresne	S
5	add	ABC123	Default Address									
6	add	nedsComp	Default Address				Ned	Finterwink	Ned's Computer Shed		1938 SW Poto	
7	add	nerdherd	Default Address				Beazus	Stankus			7365 Swatmor	
8	add	newOrlear	Default Address				Owen	Lawson	New Orleans Digital S		54 Street Aven	
9	add	northeaste	Default Address				Jason	OReilly			93767 Floatilla	
10	add	steelToolV	Default Address				Ron	Swanson	Steel Tool Werks		570 Addison St	
11	add	steelToolV	Manufacturing Floor				Rick	Grieco	Steel Tool Werks		3744 Welles A	

NOTES



For Company Name and Customer ID search types, if you know only a partial company name or customer ID, you can search by the first few characters. The search engine locates all company names or customer IDs starting with the characters entered. Wild card searches are permitted for Company Name and Customer ID search types, however, they can be time consuming to complete.

- **Customer ID** - The Customer ID search option locates records based on the customer ID field required for each Accounts record. This type of search supports partial customer IDs with a trailing wildcard, meaning the search engine returns results based on a exact match between the search criteria entered and the beginning characters of the customer ID. A search using "123A" would return records with the following IDs: "123A23", "123ABG5" and "123A-345".
- **Company Name** - The Company Name search type locates records based on the company name field used to create an Accounts record. For flexibility, this search type returns results based on partial matches. If any part of a company name matches the criteria entered, the corresponding record is included in the search results set. For example, a search for "Acme" would result in records that have any of these names: "Acme Corp", "Big Acme Inc" and "Acme Corporation".

RELATED TOPICS

ACCOUNTS

OVERVIEW

Accounts is a tool for supporting collaborative customer management in the BigMachines system. Customer records can be individually created in the Accounts database or they can be bulk uploaded into the system. Once customer records exist in Accounts, they are accessible to sales agents, channel agents and full access users.

All users that have access to customer accounts in Accounts also have permission to make updates or additions to Account records. Accounts records can contain general customer information, multiple address records for billing and shipping data, and comments. To collect more specific information, fullaccessusers can generate supplemental fields for capturing custom data. For more information, see Adding Custom Attributes.

A single Account record can be associated with one or more partner users. Associations are necessary if you plan to integrate Accounts with other areas of administration, such as configuration rules and commerce documents. To learn more about associations, see Defining Associations.

Accounts can also be integrated with commerce to auto-fill commerce document fields. Integrating Accounts with commerce speeds up document creation time by eliminating redundant manual data entry.

Managing User Accounts

The Accounts section maintains customer records. Using this page you can:

- Search for Customer Records
- Add Accounts Manually or through Integration
- Import Account Files
- Export Master Data and/or Address Information
- Create Custom Attributes for Accounts
- Customize Search: See Setup Account Search Criteria for more details.
- Customize Search Results: See Customizing Search Results for more details.

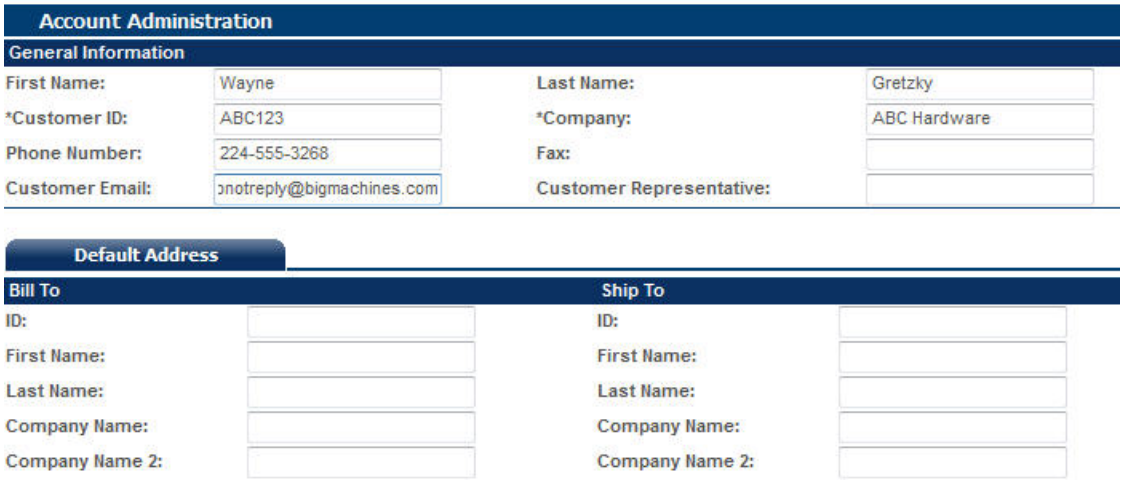
ADMINISTRATION

Creating an Account (Manually)

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Add** button.
3. Enter **General Account Information** and provide **Default Bill To/Ship To Addresses.**

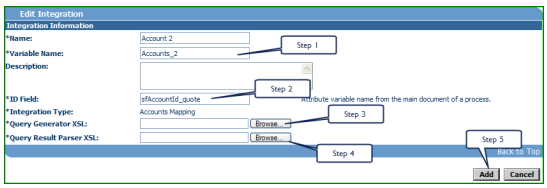
Customer ID and **Company Name** fields are required to create an Account.

4. In Additional Information, complete any custom attribute fields.
5. Click **Add**.



Creating an Account (Integration)

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click **Integration** to be taken to the Integration for CRM Page.
3. Click **Add**.
4. Enter a Name, Variable Name and Description for the new integration.
5. In the ID field, enter the attribute variable name from the main document of a process.
6. Click **Browse** to locate and add a Query Generator XSL file.
7. Click **Browse** to locate and add a Query Result Parser XSL file.
8. Click **Add** to add the new integration or click Cancel to go back to the page for managing integrations.

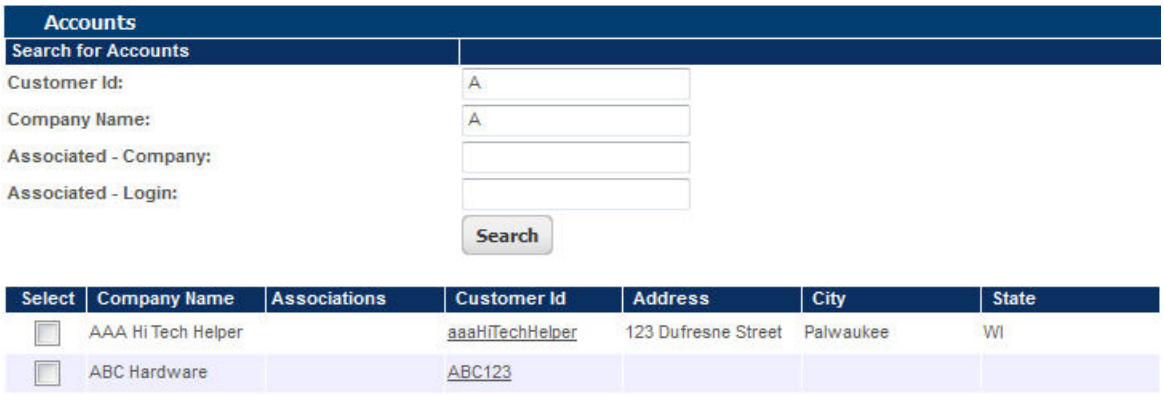


A Commerce Transaction ID field will appear after you've saved the Intgration.

9. Enter a commerce **Transaction ID** and click **Show Query Input XML** or Preview SOAP XML for additional integration details.

Managing Existing Accounts:

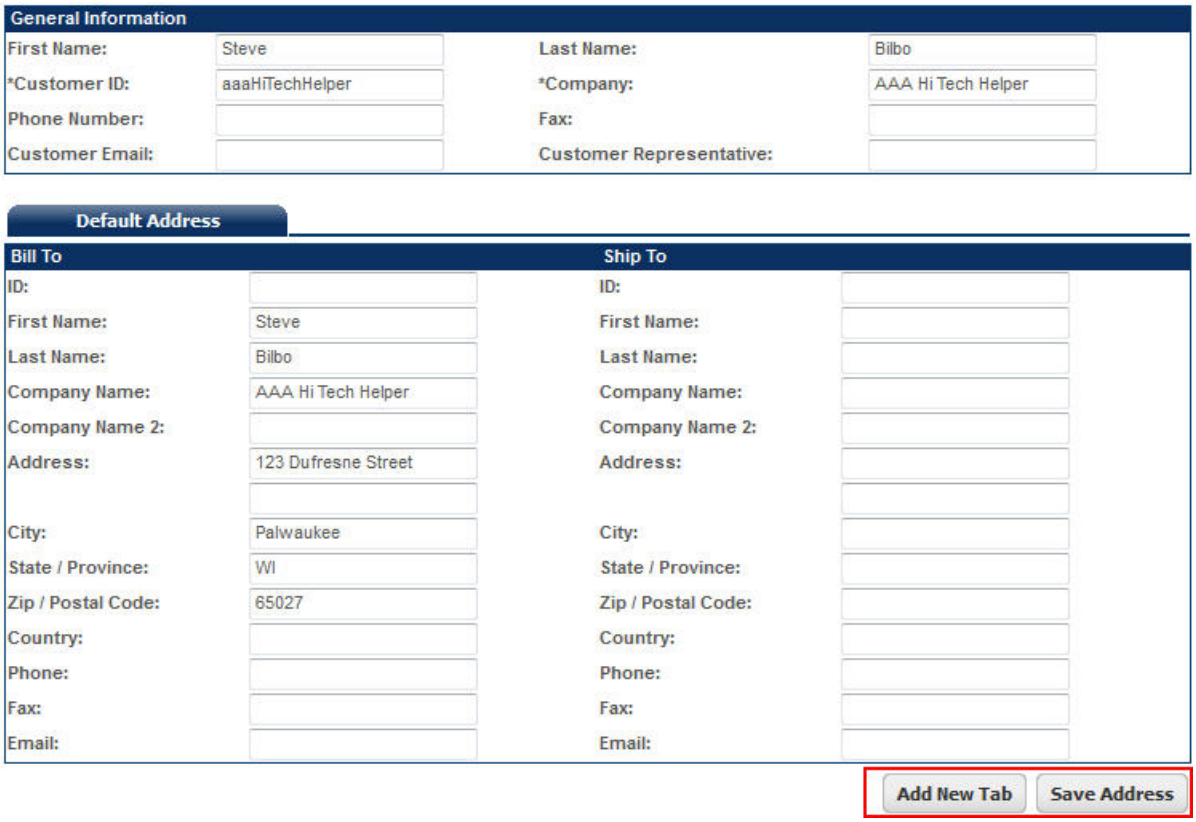
1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Search for the **Account** by Customer ID, Company Name, Associated-Company or Associated-Login.
3. Click **Search**.
4. Click on the **Customer ID**of the Account you want to edit.



5. Edit or enter account information in the fields provided.
6. Click **Add New Tab** to add an alternate address to new and existing Accounts records.
   - Enter a **tab label** for the new address, enter alternate address data, and click Save Address. Click on Delete Address when it is no longer required or valid.

     If you accidentally click the Add New Tab button instead of the Save Address button, cancel the operation. Otherwise, the new address record will be lost.

   - Alternate addresses appear as separate address tabs and can be viewed or modified by clicking the tab name.



7. Click on **Associations** to define account associations. See Defining Associations for more information.
8. Click **Update** to Save changes to information or click Back to return to the Accounts page without saving.

Importing Account Files

1. Prepare a .CSV file for Import.
2. Use the standard bulk upload actions to upload information.

> Click **Export Master Data** or **Export Address Data** for a template that you can use for the upload.

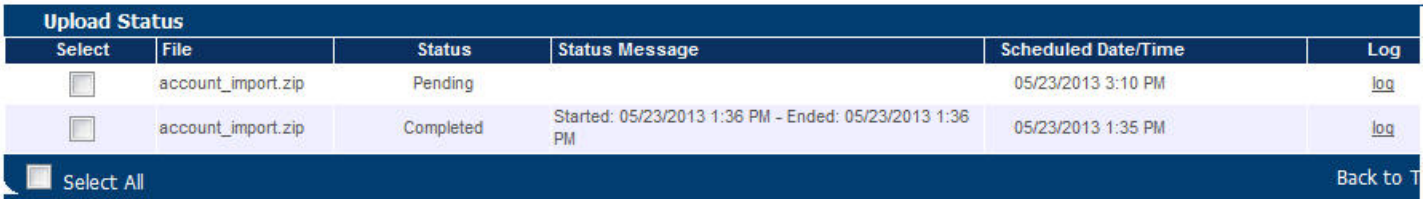
4. Enter information into the available fields. Customer ID (customer_id) and Company Name (company_name) are required.

	A	B	C	D	E	F	G	
1	Accounts Master							
2								
3	update	first_name	last_name	customer_id	company_name	customer_rep	phone	er
4	add	Steve	Bilbo	aaaHiTechHelper	AAA Hi Tech Helper			
5	add	Wayne	Gretzky	ABC123	ABC Hardware		224-555-3268	do
6	add	Ned	Finterwink	nedsComputerShed	Neds Computer Shed			
7	add	Beazus	Stankus	nerdherd	Nerd Herd			
8	add	Owen	Lawson	newOrleansDigital	New Orleans Digital			
9	add	Jason	OReilly	northeasternRack	Northeastern Rack			
10	add	Ron	Swanson	steelToolWerks	Steel Tool Werks			
11	add	Toni	Pastorino	ABC234	ABC Hardward - San Mateo		224-555-3269	do

5. Save the file and create a .zip file.
6. Navigate from Admin Home Page > Integration Platform > Accounts
7. Click the **Import** button.
8. Click the **Browse** button and choose a file.



9. Click **Add**. Repeat Steps 1-9 for all files to be imported.
10. Select the **file** and click the **Upload** button.
11. Click the **Refresh** button to see the status of the import.



12. Click the **log** to verify the upload has occurred without errors.

Export Master Data

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Export Master Data** button.

3. Select an option to Open or Save the file. The data will be downloaded as a .zip file.
4. Open the .zip file and double-click on the .csv file to open it.

	A	B	C	D	E	F	G	
1	Accounts Master							
2								
3	update	first_name	last_name	customer_id	company_name	customer_rep	phone	e
4	add	Steve	Bilbo	aaaHiTechHelper	AAA Hi Tech Helper			
5	add	Wayne	Gretzky	ABC123	ABC Hardware		224-555-3268	d
6	add	Ned	Finterwink	nedsComputerShed	Neds Computer Shed			
7	add	Beazus	Stankus	nerdherd	Nerd Herd			
8	add	Owen	Lawson	newOrleansDigital	New Orleans Digital			
9	add	Jason	OReilly	northeasternRack	Northeastern Rack			
10	add	Ron	Swanson	steelToolWerks	Steel Tool Werks			

Export Address Data

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the Export Address Data button.
3. Select an option to Open or Save the file. The data will be downloaded as a .zip file
4. Open the .zip file and double-click on the.csv file to open it.

	A	B	C	D	E	F	G	H	I	J	K	
1	Accounts A	ddresses										
2												
3	update	customer_	tab_label:	tab_label:	tab_label:	tab_label:	billing_firs	billing_last	billing_con	billing_con	billing_add	billi
4	add	aaaHiTech	Default Address				Steve	Bilbo	AAA Hi Tech Helper		123 Dufresne	S
5	add	ABC123	Default Address									
6	add	nedsComp	Default Address				Ned	Finterwink	Ned's Computer Shed		1938 SW Poto	
7	add	nerdherd	Default Address				Beazus	Stankus			7365 Swatmor	
8	add	newOrlear	Default Address				Owen	Lawson	New Orleans Digital S		54 Street Aven	
9	add	northeaste	Default Address				Jason	OReilly			93767 Floatilla	
10	add	steelToolV	Default Address				Ron	Swanson	Steel Tool Werks		570 Addison St	
11	add	steelToolV	Manufacturing Floor				Rick	Grieco	Steel Tool Werks		3744 Welles A	

NOTES



For Company Name and Customer ID search types, if you know only a partial company name or customer ID, you can search by the first few characters. The search engine locates all company names or customer IDs starting with the characters entered. Wild card searches are permitted for Company Name and Customer ID search types, however, they can be time consuming to complete.

- **Customer ID** - The Customer ID search option locates records based on the customer ID field required for each Accounts record. This type of search supports partial customer IDs with a trailing wildcard, meaning the search engine returns results based on a exact match between the search criteria entered and the beginning characters of the customer ID. A search using "123A" would return records with the following IDs: "123A23", "123ABG5" and "123A-345".
- **Company Name** - The Company Name search type locates records based on the company name field used to create an Accounts record. For flexibility, this search type returns results based on partial matches. If any part of a company name matches the criteria entered, the corresponding record is included in the search results set. For example, a search for "Acme" would result in records that have any of these names: "Acme Corp", "Big Acme Inc" and "Acme Corporation".

RELATED TOPICS

CREATE CUSTOM ATTRIBUTES FOR ACCOUNTS

OVERVIEW

You can customize account attributes that can be used in both the Account Search and the Search Results section. These attributes are also added to the Additional Information section that is available when creating or managing accounts.

Within Accounts, there are a set of default fields for gathering customer information. The default fields define the extent of customer data your company can collect, including general information, like customer name and ID, plus billing and shipping data and customer comments. To expand the usefulness of Accounts, fullaccessusers can customize the Accounts input form with supplemental fields. Such fields appear in addition to default fields and enable your company to collect and share other types of customer data. After custom fields are administrated, they become visible on all customer records in the Information section at the bottom of the Customer Profile page. This page enables you to manage custom attributes. These attributes appear on the Account Profile page.

ADMINISTRATION

Creating Custom Attributes for Accounts

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Customize Attributes** button.
3. Click **Add** on the Accounts Custom Attribute List page.
4. Enter a **Name** and **Variable Name** for the attribute.
5. Choose the **Field Type** from the drop-down menu: String, Single Select Menu, or multi-select Menu.
6. Click **Add**.
7. Enter the values for the **available fields** based on data and field type:

Element	Field Type	Description
Description	All available	Enter details for the attribute.
Default Value	All Available	Enter a default value that will pre-populate the custom attribute.
Required	All available	Select if the field must be populated when creating and/or managing an account.
Show Field	String and Single Select Menu.	Select Search Page to use the attribute in an Account Search. Select Search Results Page to use the attribute as a field in the Account Search Results.
Menu Population	Single Select Menu and multi-select Menu	Populate menu values. Click here for more information.
Filter Option	Single Select Menu	Use the Filter Option section to specify whether or not you want to add a filter script. If you select Define Filter Script, you need to click Define Function. This will display the function editor window for creating that script.

8. Click **Update**.
9. Repeat Steps 1 - 8 for all custom attributes.

10. Select **one or more** custom attributes and click **Deploy**.
11. Click **Back** to return to the Accounts page.



Notice that the attributes where Search Page is selected are now displayed.

12. To Delete Custom Attributes:
    - Select custom attributes that are to be deleted and click Delete to remove the fields. The refreshed view of the page shows the deletion.



When custom fields are deleted, the associated data for the field gets deleted too. There is no way to recover deleted data. To save custom field data, you can edit fields instead of deleting them.

13. To re-order custom attributes and create a new order:
    - Enter a new numerical sequence under the Order column. The ordering system uses 1 as the highest order number.
14. Click **Update** to make the changes permanent and return to the Accounts page or click Back to return without saving changes.

RELATED TOPICS

DEFINING ASSOCIATIONS FOR ACCOUNTS

OVERVIEW

A single Accounts record can be associated with one or more user accounts. The purpose of associations is that they enable you to support Accounts-based functionality. Whenever the system recognizes an association between a user and an Account record, it can trigger Accounts-based functionality. For example, this would allow configuration rules to fire when the associated Account satisfies a rule's condition.

There are two types of association:

- **Individual:** Associate an individual user account to a single Accounts customer record.
- **Company:** Define associations for buying companies using an alias called "_everyone@buyercompany".

When a user logs in, the system looks for an individual association. If one exists, then that informs the system which record to associate with the user. If an individual association does not exist, the system looks for a company association for the user's company. If a company association exists, then that informs the system which record to associate with the user. If neither an individual nor a company association exists for a user, then the user is not associated with any Account record and Accounts-based functionality will not trigger.

ADMINISTRATION

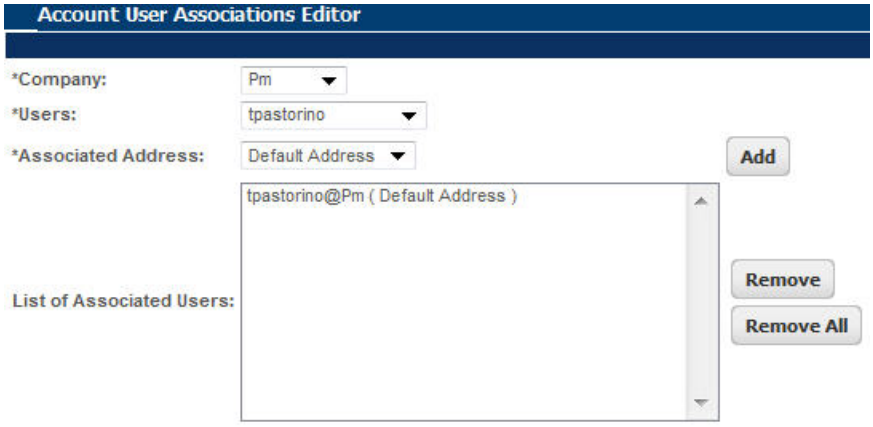
Adding an Association

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Search for an **account** and select its **Customer ID** or create a new account.
3. Click the **Associations** button.
4. Choose the **Company** to associate your Account to from the drop-down menu.



The companies that will be available are the host company and any Buy Access Partner Organizations.

5. Choose an **individual user** you would like to Associate from the Users drop-down menu.
6. Choose an **address** from the Associated Addresses drop-down menu.



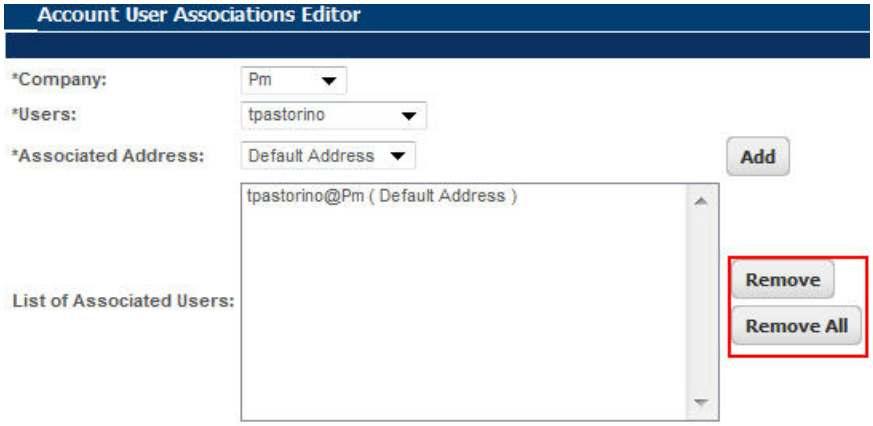
7. Click **Add**.
8. Repeat Steps 5-7 for all individual users.
9. Click **Update** to save changes and return to the main page. You will see the Users associated with the Account now appear under Associations for the Account.

Individual users accounts can only be associated with a single Account record. Similarly, company wide associations can only be defined for a single Account. New associations take effect immediately after clicking Update.

Removing an Association

In order to associate a user account with an Account record, the user account cannot be associated with any other record. Therefore, in order to change an association, any existing association must be removed before administrating a new one.

1. Highlight an **account** and click **Remove**.
2. To remove all associations: Click **Remove All**.



3. Click **Update** to save association changes.

NOTES

Associations can be managed by FullAccess users only. After associations are created, channel agents, sales agents and full access users can conduct Account searches using them.

RELATED TOPICS

SETUP ACCOUNT SEARCH CRITERIA

OVERVIEW

There is an option in Accounts that allows you to select fields to use when searching for Accounts. This list only includes standard account attributes. You can not add custom attributes from here.



ADMINISTRATION

1. Navigate from Admin Home Page > Integration Platform > Accounts
2. Click the **Customize Search** button.
3. Click **Add**.
4. Select the **account attribute(s)** you want to add click **Add**.



5. Specify an **order number** for the attributes.
6. Change **Search Attribute Labels**: Type the new label in the appropriate Label field and click Update Delete Search Attributes:
7. Click **Translations** to include translations for each label in all languages which your application supports.
8. Click **Update** to make the changes permanent.

9. Click **Back** to go back to the customizing search results page.



RELATED TOPICS

CUSTOMIZE SEARCH RESULTS FOR ACCOUNTS

OVERVIEW

You can customize the way you want to view the search results that are displayed on the Accounts page.

Select	Company Name	Associations	Customer Id	Address	City	State	Last Name	First Name
☐	AAA Hi Tech Helper		aaaHiTechHelper	123 Dufresne Street	Palwaukee	WI	Bilbo	Steve
☐	ABC Hardware		ABC123				Gretzky	Wayne
☐	ABC Hardward - San Mateo		ABC234				Pastorino	Toni

ADMINISTRATION

Customizing Search Results

1. Navigation from the Admin Home Page > Integration Platform > Accounts
2. Click **Customize Search Results**.
3. Click **Add** to add attributes to the Account Search Results Attributes List page.
4. Select the **account attribute(s)** you want to add click **Add**.
5. Specify an **order number** for the attributes.
6. Change **Search Results Attribute Labels**: Type the new label in the appropriate Label field and click Update Delete Search Attributes:

Accounts Search Result Attributes List

Select	Order	Attribute Label	Label
☐	1	Company Name	Company Name
☐	2	Associated - Login	Associations
	3	Customer Id	Customer Id
☐	4	Billing Address	Address
☐	5	Billing City	City
☐	6	Billing State	State
☐	7	Last Name	Last Name
☐	8	First Name	First Name

7. Click **Translations** to include translations for each label in all languages which your application supports.
8. Click **Update** to make the changes permanent.
9. Click **Back** to go back to the customizing search results page.

RELATED TOPICS

WSDL OVERVIEW

OVERVIEW

To support a flexible business model, there is a need to provide a way to generate the interface for SOAP interaction in a dynamic manner. This will allow the client to define and make changes to their product, configuration and commerce details and easily generate an interface reflecting the latest changes. To suffice the above requirement, a programmatic way to generate the interface description is needed. The generated WSDL can then be used to generate a SOAP message using standard open source tools including ApacheWSDL2Java. The soap interaction can be automated using the generated WSDL.

- Based on existing SOAP message the WSDL currently will support only the RPC style SOAP interaction.
- Since the current SOAP message defines its own structure and data types literal encoding will be followed to create the SOAP message on our defined schema.
- As of now since interaction is limited only to SOAP over HTTP, the current requirement is to provide the SOAP interaction with support for HTTP as the transport protocol.

Required Fields

BigMachines has 9 WSDLs that define the web services and how to access them. Each WSDL has required fields that must be inputted into the request. There are also required fields that are returned when a web service has successfully processed and when web service is unsuccessful. There are other fields that can be inputted or returned but are not required to have a successfully processed web service or be populated in the response back from the web service. The fields listed below are the ones that are marked as required by the WSDL. These are fields that must always be populated or are returned after a web service is executed.

ADMINISTRATION

Types of WSDL

WSDL	Description	Notes
Generic	Basic description of interface common across clients	The generic WSDL is based on the abstract interface design which independent of any configuration done by fullaccessuser of a particular client. Thus this WSDL remains common across all the clients and represents the basic structure of the SOAP message which can be used for interaction between SOAP server and the client. The SOAP message generated based on this WSDL cannot be used directly to perform any operation but is useful for understanding the basic SOAP message structure to be used and explaining the client's SOAP functionality.
Specific	Interface description used to send SOAP message and varies as per the changes by fullaccessuser.	The specific WSDL represents the WSDL, which varies for each client based on their own specified set up including parts, commerce and other modules. This WSDL is used for generating the SOAP message and communication between the SOAP server and the client. Also whenever some changes are affected by the admin this WSDL is generated again for reflecting the latest changes.

WSDL per Module

Whenever you makes changes to specific set-ups to the above-mentioned areas, this WSDL must be regenerated. Look at the table below to understand WSDL regeneration for individual sections.

Section	WSDL Information
Security	Since the SOAP XML for these operations remains the same independent of changes done by fullaccessuser the high level and concrete WSDL are same for Security Service. WSDL is available by default. It has to be regenerated only when a company's base information is modified (e.g. supported languages/ currencies etc.)
Commerce	**Generic WSDL:** The high level WSDL will be made independent of any defined processes and therefore the <document> and <attribute> elements within it, will not be defined. Since other elements are common for all processes and already defined in the BM system the WSDL should generate the SOAP message without the document attributes. **Specific WSDL:** The concrete WSDL is process specific, to specify the document attributes of various documents and sub documents defined by fullaccessuser in a commerce process. Thus one concrete WSDL will be generated per commerce process defined. This WSDL is automatically generated every time a commerce process is deployed.
Configuration	Following the commerce approach, configuration has a common high level and concrete WSDL. For deployed segments, the WSDL is automatically generated at the time of deployment. For the non-deployable segments, the WSDL can be generated manually from the web services section. Note: For deployable segments, WSDL cannot be generated from the 'Generate WSDL' page.
Parts	**Generic WSDL:** This WSDL being independent of any administration set up, will show all the available fields of custom attributes irrespective of whether they are used or not. This way WSDL remains independent of any given configuration and can be used to generate a SOAP message, since custom fields not used are not considered. **Specific WSDL:** The concrete WSDL will change based on number of custom attributes defined by each fullaccessuser. Therefore each time the fullaccessuser modifies and deploys custom attributes, the WSDL will be regenerated. This WSDL will only define the custom attributes that are used. The WSDL will also specify the properties of the custom attributes as defined by the fullaccessuser e.g. whether this attribute is required, if it's a menu type item etc.
Price Book Associations	WSDL is available by default. It has to be regenerated only when a company's base information is modified (for example, supported languages/ currencies etc.). Generic and Specific WSDL are the same for Price Book Associations.
Data Tables	**Generic WSDL:** The high level or generic WSDL will be common across all different tables. To retain the generic property of this WSDL, all the default DB column names of BM_SCRIPT_TABLE will be made XML elements. This makes the WSDL independent of any table and describes the basic SOAP message structure of different operations of Data Tables Service. **Specific WSDL:** For each table, a different WSDL needs to be generated representing the specific columns of a given table. WSDL is automatically generated whenever the table definition is modified. If a table does not have any rows, a row must be added and the table definition updated, before the WSDL can be generated. The current SOAP message structure will be similar to the SOAP request and responses generated from a given concrete WSDL of a given table.
Groups	
Users	
Exchange Rates	

RELATED TOPICS

AN OVERVIEW OF SOAP API

OVERVIEW

The system allows the fullaccessuser to setup all their products, configuration, and commerce processes details. The fullaccessuser is able to define and change details of their products, rules to be applied on configuring a product and finally create and update their commerce processes, which form the basis of a particular commerce transaction. An application wanting to consume services can use SOAP messages to interact with BigMachines and take advantage of its features.

The BigMachines system supports SOAP API for the purpose of uploading large amounts of system data. System data can be sent through a client program to the BigMachines system.

For sending information to BigMachines system:

1. Data is sent from a client program to BigMachines API server and modify data on your system.
2. Client program should get the result of the API call with a status field, including pass or fail.

Using SOAP API, you can perform many useful tasks, such as updating the parts and commerce information. You don't have to use the FTP clients or BigMachines GUI to upload and download data.

ADMINISTRATION

Session ID Authentication

BigMachines offers sophisticated category identification and session id authentication for all SOAP APIs. The SOAP Header requires a category element to identify the module, including commerce, security, data tables etc, to which the API call is being made. There is also a session id validation mechanism providing enhanced security to API calls.

- SOAP Header - The SOAP Header contains the <bm:userInfo> tag, which is required for all actions.
- Session Id - With version 7.0.0 of the BigMachines , enhanced security is adopted for SOAP API services by providing a session id. Once a user logs in using the API, a session id is generated and returned to the user. For all subsequent API calls, the action xml sent to the SOAP server must contain the <bm:userInfo><sessionId> tag, with the system generated session id.

SOAP calls are supported in the following modules

- Security API
- Commerce API
- Configuration API
- Parts API
- Price Book Associations API
- Data Tables API
- Users API

- Groups API
- Exchange Rates API

RELATED TOPICS

WEB SERVICES

OVERVIEW

Web Services refers to a system of communication for diverse applications talking over the World Wide Web. By using Web services, the application can publish its function or message to the rest of the world.

Typical components of a web service are:

- Extensible Markup Language (XML) constituting the data to be exchanged.
- World Wide Web's Hyper Text Transfer Protocol (HTTP) constituting the network over which data is to be exchanged.
- Simple Object Access Protocol (SOAP) constituting the message protocol so applications understand one another.

SOAP is a simple XML-based communication protocol to let applications exchange information over HTTP. It is used to accomplish communication between diverse applications independent of the platform, technology, and programming languages. SOAP is a World Wide Web Consortium (W3C) standard.

Web Services Description Language (WSDL) is an XML document that describes a web service. It can also be defined as a standard to describe an interface for exchanging information using the SOAP protocol. This interface describes the location of the SOAP service, nature of messages exchanged, in addition to their structure, elements, and characteristics like null allowed, minimum occurrences etc.

ADMINISTRATION

To learn more about these technologies please refer to the table below:

Technology	Information/Tutorial URL
XML	www.w3schools.com/xml/xml_whatis.asp
SOAP	www.w3schools.com/soap/default.asp
WSDL	www.w3schools.com/wsdl/default.asp
W3C	www.w3c.org

RELATED TOPICS

WSDL OVERVIEW

OVERVIEW

To support a flexible business model, there is a need to provide a way to generate the interface for SOAP interaction in a dynamic manner. This will allow the client to define and make changes to their product, configuration and commerce details and easily generate an interface reflecting the latest changes. To suffice the above requirement, a programmatic way to generate the interface description is needed. The generated WSDL can then be used to generate a SOAP message using standard open source tools including ApacheWSDL2Java. The soap interaction can be automated using the generated WSDL.

- Based on existing SOAP message the WSDL currently will support only the RPC style SOAP interaction.
- Since the current SOAP message defines its own structure and data types literal encoding will be followed to create the SOAP message on our defined schema.
- As of now since interaction is limited only to SOAP over HTTP, the current requirement is to provide the SOAP interaction with support for HTTP as the transport protocol.

Required Fields

BigMachines has 9 WSDLs that define the web services and how to access them. Each WSDL has required fields that must be inputted into the request. There are also required fields that are returned when a web service has successfully processed and when web service is unsuccessful. There are other fields that can be inputted or returned but are not required to have a successfully processed web service or be populated in the response back from the web service. The fields listed below are the ones that are marked as required by the WSDL. These are fields that must always be populated or are returned after a web service is executed.

ADMINISTRATION

Types of WSDL

WSDL	Description	Notes
Generic	Basic description of interface common across clients	The generic WSDL is based on the abstract interface design which independent of any configuration done by fullaccessuser of a particular client. Thus this WSDL remains common across all the clients and represents the basic structure of the SOAP message which can be used for interaction between SOAP server and the client. The SOAP message generated based on this WSDL cannot be used directly to perform any operation but is useful for understanding the basic SOAP message structure to be used and explaining the client's SOAP functionality.
Specific	Interface description used to send SOAP message and varies as per the changes by fullaccessuser.	The specific WSDL represents the WSDL, which varies for each client based on their own specified set up including parts, commerce and other modules. This WSDL is used for generating the SOAP message and communication between the SOAP server and the client. Also whenever some changes are affected by the admin this WSDL is generated again for reflecting the latest changes.

WSDL per Module

Whenever you makes changes to specific set-ups to the above-mentioned areas, this WSDL must be regenerated. Look at the table below to understand WSDL regeneration for individual sections.

Section	WSDL Information
Security	Since the SOAP XML for these operations remains the same independent of changes done by fullaccessuser the high level and concrete WSDL are same for Security Service. WSDL is available by default. It has to be regenerated only when a company's base information is modified (e.g. supported languages/ currencies etc.)
Commerce	**Generic WSDL:** The high level WSDL will be made independent of any defined processes and therefore the <document> and <attribute> elements within it, will not be defined. Since other elements are common for all processes and already defined in the BM system the WSDL should generate the SOAP message without the document attributes. **Specific WSDL:** The concrete WSDL is process specific, to specify the document attributes of various documents and sub documents defined by fullaccessuser in a commerce process. Thus one concrete WSDL will be generated per commerce process defined. This WSDL is automatically generated every time a commerce process is deployed.
Configuration	Following the commerce approach, configuration has a common high level and concrete WSDL. For deployed segments, the WSDL is automatically generated at the time of deployment. For the non-deployable segments, the WSDL can be generated manually from the web services section. Note: For deployable segments, WSDL cannot be generated from the 'Generate WSDL' page.
Parts	**Generic WSDL:** This WSDL being independent of any administration set up, will show all the available fields of custom attributes irrespective of whether they are used or not. This way WSDL remains independent of any given configuration and can be used to generate a SOAP message, since custom fields not used are not considered. **Specific WSDL:** The concrete WSDL will change based on number of custom attributes defined by each fullaccessuser. Therefore each time the fullaccessuser modifies and deploys custom attributes, the WSDL will be regenerated. This WSDL will only define the custom attributes that are used. The WSDL will also specify the properties of the custom attributes as defined by the fullaccessuser e.g. whether this attribute is required, if it's a menu type item etc.
Price Book Associations	WSDL is available by default. It has to be regenerated only when a company's base information is modified (for example, supported languages/ currencies etc.). Generic and Specific WSDL are the same for Price Book Associations.
Data Tables	**Generic WSDL:** The high level or generic WSDL will be common across all different tables. To retain the generic property of this WSDL, all the default DB column names of BM_SCRIPT_TABLE will be made XML elements. This makes the WSDL independent of any table and describes the basic SOAP message structure of different operations of Data Tables Service. **Specific WSDL:** For each table, a different WSDL needs to be generated representing the specific columns of a given table. WSDL is automatically generated whenever the table definition is modified. If a table does not have any rows, a row must be added and the table definition updated, before the WSDL can be generated. The current SOAP message structure will be similar to the SOAP request and responses generated from a given concrete WSDL of a given table.
Groups	
Users	
Exchange Rates	

RELATED TOPICS

SOAP PARTS API

OVERVIEW

You can generate a skeleton input for each operation of the parts WSDL, which contains 4 operations:

- addParts
- updateParts
- getParts
- deleteParts

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category: Security
- schemaLocation

You can add, update, delete or get parts based on the SOAP interfaces provided. This will help in real-time integration with the parts database so as to keep data in synch. The following SOAP API will be provided for enabling integration with the parts section:

View API Details

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
addParts	Adds a part. It expects input of all the part details and returns the status whether the part was successfully added or not.	• sessionId • category • schemaLocation • part_number	**status:** • success • message	exceptionCode exceptionMessage
getParts	Returns the part details of parts which are searched based on the given criteria (one or multiple).	• sessionId • category • schemaLocation • field • value • comparator	**status:** • success • message **parts** • eachRecord • part_number	exceptionCode exceptionMessage
updateParts	Update the details of the specified part number. The return specifies whether the operation was successful.	• sessionID • category • schemaLocation • part_number	**status:** • success • message	exceptionCode exceptionMessage
deleteParts	Deletes a specific part. The	• sessionId	**status:**	exceptionCode

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
	part(s) deletes are based on one or more specified criteria.	• category • schemaLocation • field • value • comparator	• success • message	exceptionMessage

ADMINISTRATION

Accessing User WSDL and generating skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Parts** tab.
3. Select one of **parts operations** from the Select API drop-down menu.
4. Enter a **Part Number**.
5. Click **Generate Input** to retrieve the skeleton.

USE CASES

Examples of Parts API:

addParts - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>BADEA7F1C53874276177D250E729DCA5</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Parts</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/parts/Parts.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:addParts xmlns:bm="urn:soap.bigmachines.com">
<bm:Parts>
<bm:each_record>
<bm:part_number>RAM-128</bm:part_number>
```

```
<bm:direct_buy>1</bm:direct_buy>
<bm:description>
<bm:en>128 GB RAM 1333MHz</bm:en>
<bm:nl_NL/>
<bm:de/>
<bm:ja_JP/>
</bm:description>
<bm:units>
<bm:en/>
<bm:nl_NL/>
<bm:de/>
<bm:ja_JP/>
</bm:units>
<bm:lead_time>-1</bm:lead_time>
<bm:_part_custom_field1>Memory</bm:_part_custom_field1>
<bm:_part_custom_field2>840</bm:_part_custom_field2>
<bm:_part_custom_field3>5</bm:_part_custom_field3>
<bm:_part_custom_field4/>
<bm:_part_custom_field5>1400.0</bm:_part_custom_field5>
<bm:_part_custom_field6>0.0</bm:_part_custom_field6>
<bm:_part_custom_field8>Hardware</bm:_part_custom_field8>
<bm:partner_part_id/>
<bm:partner_std_pbook_entry_id/>
<bm:last_integration_time/>
<bm:price>
<bm:USD>1400</bm:USD>
<bm:EUR>1076</bm:EUR>
<bm:GBP>904</bm:GBP>
<bm:AUD>1409</bm:AUD>
<bm:JPY>109017</bm:JPY>
<bm:ESP/>
</bm:price>
</bm:each_record>
</bm:Parts>
</bm:addParts>
</soapenv:Body>
</soapenv:Envelope>
```

---

updateParts - Input SOAP XML

<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>BADEA7F1C53874276177D250E729DCA5</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Parts</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/parts/Parts.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:updateParts xmlns:bm="urn:soap.bigmachines.com">
<bm:Parts>
<bm:each_record>
<bm:part_number>RAM-128</bm:part_number>
<bm:direct_buy>1</bm:direct_buy>
<bm:description>
<bm:en>128 GB RAM 1333MHz</bm:en>
<bm:nl_NL/>
<bm:de/>
<bm:ja_JP/>
</bm:description>
<bm:units>
<bm:en/>
<bm:nl_NL/>
<bm:de/>
<bm:ja_JP/>
</bm:units>
<bm:lead_time>-1</bm:lead_time>
<bm:_part_custom_field1>Memory</bm:_part_custom_field1>
<bm:_part_custom_field2>840</bm:_part_custom_field2>
<bm:_part_custom_field3>5</bm:_part_custom_field3>
<bm:_part_custom_field4/>
<bm:_part_custom_field5>1400.0</bm:_part_custom_field5>
<bm:_part_custom_field6>0.0</bm:_part_custom_field6>
<bm:_part_custom_field8>Hardware</bm:_part_custom_field8>
<bm:partner_part_id/>
<bm:partner_std_pbook_entry_id/>
<bm:last_integration_time/>

```
<bm:price>
<bm:USD>1400</bm:USD>
<bm:EUR>1076</bm:EUR>
<bm:GBP>904</bm:GBP>
<bm:AUD>1409</bm:AUD>
<bm:JPY>109017</bm:JPY>
<bm:ESP/>
</bm:price>
</bm:each_record>
</bm:Parts>
</bm:updateParts>
</soapenv:Body>
</soapenv:Envelope>
```

---

getParts - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>BADEA7F1C53874276177D250E729DCA5</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Parts</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/parts/Parts.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getParts xmlns:bm="urn:soap.bigmachines.com">
<bm:Parts>
<bm:criteria>
<bm:field>part_number</bm:field>
<bm:value>RAM-128</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>direct_buy</bm:field>
<bm:value>1</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
```

```
<bm:criteria>
<bm:field>description</bm:field>
<bm:value>128 GB RAM 1333MHz</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>units</bm:field>
<bm:value/>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_added</bm:field>
<bm:value>2010-11-23 11:06:24</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_modified</bm:field>
<bm:value>2012-03-24 05:11:45</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>lead_time</bm:field>
<bm:value>-1</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>_part_custom_field1</bm:field>
<bm:value>Memory</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>_part_custom_field2</bm:field>
<bm:value>840</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>_part_custom_field3</bm:field>
<bm:value>5</bm:value>
<bm:comparator>=</bm:comparator>
```

```
</bm:criteria>
<bm:criteria>
<bm:field>_part_custom_field4</bm:field>
<bm:value/>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>_part_custom_field5</bm:field>
<bm:value>1400.0</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>_part_custom_field6</bm:field>
<bm:value>0.0</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>_part_custom_field8</bm:field>
<bm:value>Hardware</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>partner_part_id</bm:field>
<bm:value/>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>partner_std_pbook_entry_id</bm:field>
<bm:value/>
<bm:comparator>=</bm:comparator>
</bm:criteria>
</bm:Parts>
</bm:getParts>
</soapenv:Body>
</soapenv:Envelope>
```

---

deleteParts - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>BADEA7F1C53874276177D250E729DCA5</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Parts</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/parts/Parts.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:deleteParts xmlns:bm="urn:soap.bigmachines.com">
<bm:Parts>
<bm:criteria>
<bm:field>part_number</bm:field>
<bm:value>RAM-128</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>direct_buy</bm:field>
<bm:value>1</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>description</bm:field>
<bm:value>128 GB RAM 1333MHz</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>units</bm:field>
<bm:value/>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_added</bm:field>
<bm:value>2010-11-23 11:06:24</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_modified</bm:field>
```

<bm:value>2012-03-24 05:11:45</bm:value>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

<bm:field>lead_time</bm:field>

<bm:value>-1</bm:value>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

<bm:field>_part_custom_field1</bm:field>

<bm:value>Memory</bm:value>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

<bm:field>_part_custom_field2</bm:field>

<bm:value>840</bm:value>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

<bm:field>_part_custom_field3</bm:field>

<bm:value>5</bm:value>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

<bm:field>_part_custom_field4</bm:field>

<bm:value/>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

<bm:field>_part_custom_field5</bm:field>

<bm:value>1400.0</bm:value>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

<bm:field>_part_custom_field6</bm:field>

<bm:value>0.0</bm:value>

<bm:comparator>=</bm:comparator>

</bm:criteria>

<bm:criteria>

```
<bm:field>_part_custom_field8</bm:field>
<bm:value>Hardware</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>partner_part_id</bm:field>
<bm:value/>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>partner_std_pbook_entry_id</bm:field>
<bm:value/>
<bm:comparator>=</bm:comparator>
</bm:criteria>
</bm:Parts>
</bm:deleteParts>
</soapenv:Body>
</soapenv:Envelope>
```

RELATED TOPICS

SOAP USERS API

OVERVIEW

You can generate a skeleton input for each operation in the users WSDL, which contains 4 operations:

- addUsers
- updateUsers
- getUser
- resetPassword

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category:Users
- schemaLocation

View API Details

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
addUsers	Adds a user through SOAP. When multiple users are created using this method, the entire request is processed before a response is sent.	sessionID category schemaLocation login company_name type first_name email super_user_access_perm unplugged_default_check_in separate_ship_addr is_notify_email **Note**: The above fields are the minimum requirement to add a user. The following fields will be set with default values if not set in the web services API. bm_language: English	**status:** success message **detailedStatus:** record	exceptionCode exceptionMessage

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
		currency_preference: US Dollar number_format: ####.## date_format: MM/dd/yyy h:mm time_zone: (GMT-5:00 GMT-4:00) US Eastern Time		
updateUsers	Updates user information through SOAP.	sessionID category schemaLocation login super_user_access_perm unplugged_default_check_in separate_ship_addr is_notify_email	**status:** success message **detailedStatus:** record	exceptionCode exceptionMessage
getUser	Grabs user information through SOAP.	sessionID category schemaLocation login	**status:** success message **userInfo:** login first_name variable_name	exceptionCode exceptionMessage
resetPasswords	Resets user passwords through SOAP.	sessionID category schemaLocation login	**status:** success message	exceptionCode exceptionMessage

ADMINISTRATION

Accessing User WSDL and generating a skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Users** tab.
3. Select one of **users operations** from the Select API drop-down menu.
4. Enter a **User Login**.
5. Click **Generate Input** to retrieve the skeleton.

USE CASES

Examples of Users API:

addUsers - Input SOAP XML

You'll need to use a generic user login, or even your own, to generate the input XML. You can replace the username, password, and other user details.

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Users</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/users/Users.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:addUsers xmlns:bm="urn:soap.bigmachines.com">
<bm:User>
<bm:each_record>
<bm:company_name>PM</bm:company_name>
<bm:login>jimmyuser</bm:login>
<bm:password>testpass</bm:password>
<bm:type>FullAccess</bm:type>
<bm:first_name>Jimmy</bm:first_name>
<bm:last_name>User</bm:last_name>
<bm:job_title/>
<bm:email>juser@bigmachines.com</bm:email>
</bm:User>
</bm:addUsers>
</soapenv:Body>
</soapenv:Envelope>
```

---

updateUsers - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
```

```
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Users</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/users/Users.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:updateUsers xmlns:bm="urn:soap.bigmachines.com">
<bm:User>
<bm:each_record>
<bm:company_name>PM</bm:company_name>
<bm:login>jimmyuser</bm:login>
<bm:password>temppass</bm:password>
<!--The rest of the user records is the same as the addUsers API shown above -->
</bm:each_record>
</bm:User>
</bm:updateUsers>
</soapenv:Body>
</soapenv:Envelope>
```

---

getUser - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Users</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v10/users/Users.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getUser xmlns:bm="urn:soap.bigmachines.com">
<bm:userInfo>
<bm:login>jimmyuser</bm:login>
</bm:userInfo>
</bm:getUser>
</soapenv:Body>
</soapenv:Envelope>
```

resetPasswords - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Users</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v10/users/Users.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:resetPasswords xmlns:bm="urn:soap.bigmachines.com">
<bm:userInfo>
<bm:login>jimmyuser</bm:login>
</bm:userInfo>
<bm:userInfo>
<bm:login>jimmyuser2</bm:login>
</bm:userInfo>
</bm:resetPasswords>
</soapenv:Body>
</soapenv:Envelope>
```



Highlighted tags are required.

RELATED TOPICS

SOAP COMMERCE API

OVERVIEW

You can generate a skeleton input for each operation of the commerce WSDL, which contains 7 operations:

- createTransaction
- addToTransaction
- removeFromTransaction
- updateTransaction
- getTransaction
- exportFileAttachments
- importFileAttachments

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category: Security
- schemaLocation

The commerce APIs provide functionality that is available through the regular bulk upload of the Asset Manager or the Real Time API provided through WebMethods. In addition, they will provide the ability to create a new transaction (both with and without line items).

Supported API Details

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
createTransaction	Supports the create of a commerce transaction without line items, as well as transactions with non-configurable line items from a specified process. For all transactions that need to be created with configurable line items, the configuration SOAP API will have to be invoked. This will return the Transaction ID.	sessionID category schemaLocation process_var_name var_name attribute	**status:** success message **transactionID** **trasnaction** action_date	exceptionCode exceptionMessage
addToTransaction	Adds a new items to an existing transaction. It performs the "Add from Catalog" action. The request message specifies the process, document and action on the document, along with the items to be	transactionID sessionID	**status:** success	exceptionCode excep-

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
	added as parameters of the operation.	category schemaLocation process_var_name var_name attribute document_number document_var_name **To add a part:** All input variables above part quantity price_book_var_name	message **transaction** action_data	tionMessage
removeFromTransaction	Removes an item from an existing transaction. It performs the "Remove Line Items" action. The input parameters include the document ID, document number, process name and document name. They are wrapped in a specific structure described below.	transaction ID sessionID category schemaLocation process_var_name document_number document_var_name	**status:** success message	exceptionCode exceptionMessage
updateTransaction	Updates an existing transaction. It performs all "Modify" type actions, as well as the "Auto-fill" action. It does NOT support the 'Create Document' action.	transactionID sessionID category schemaLocation id process_var_name	**status:** success message	exceptionCode exceptionMessage
getTransaction	Returns the transaction based on the transaction ID, which is provided as input parameter. The API returns the complete transaction XML.	transactionID sessionID category schemaLocation id var_name attribute	**status:** success message **transaction** action_data	exceptionCode exceptionMessage
exportFileAttachment	Export a file attachment using one of two methods to stream the data via SOAP: inline base64 content in a SOAP message or binary stream with MIME containers via MTOM transmission. These meth-	sessionID category	**status:** success message	exceptionCode exceptionMessage

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
	ods provide read/write of multiple attributes at once per transaction. This API can only be used by FullAccess users with Permission to Modify Users. There are two modes available for use: Content: Retrieves the content of the attached field(s). Metadata: Retrieves the file information or metatdata for the references fields.	schemaLocation mode document_number variable_name process_var_name id	**id** **attachments**	
importFileAttachment	Import a file attachment using one of two methods to stream the data via SOAP: inline base64 content in a SOAP message or binary stream with MIME containers via MTOM transmission. These methods provide read/write of multiple attributes at once per transaction. This API can only be used by FullAccess users with Permission to Modify Users. There are two modes available for use: Update: Attaches a files, or a set of files, to the transaction. Delete: Removes a file from the file attachment attribute.	sessionID category schemaLocation mode document_number variable_name filename process_var_name id	**status:** success message	exceptionCode exceptionMessage

Returning Specific Attributes in Commerce APIs

Return specific attributes from a web service request with the <bm:return_specific_attributes /> tag. For example, you can pull in four main-document attributes and two sub-document attributes and a shortened version of the transaction XML, displaying only those attributes, will be returned. The addToTransaction, createTransaction and getTransaction API's can return specific attributes.

ADMINISTRATION

Accessing Users WSDL and generating the skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Commerce** tab.
3. Choose from the **Select API** drop-down menu.
4. Use the **Process Name** drop-down menu to choose a commerce process.
5. Click **Generate Input** to retrieve the skeleton.

USE CASES

Examples of Users API:

createTransaction - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:createTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:items>
<bm:catalogItem>
<bm:segment/>
<bm:product_line/>
<bm:model/>
</bm:catalogItem>
<bm:partItem>
<bm:part/>
<bm:quantity>1</bm:quantity>
<bm:price_book_var_name/>
</bm:partItem>
</bm:items>
<bm:transaction>
<bm:process_var_name>quickstart_commerce_process_bmClone_1</bm:process_var_name>
<bm:_bm_cm_new_transaction_currency/>
<bm:return_specific_attributes>
<bm:documents>
<bm:document>
<bm:var_name>quote_process</bm:var_name>
<bm:attributes>
<bm:attribute>_document_number</bm:attribute>
</bm:attributes>
</bm:document>
</bm:documents>
</bm:return_specific_attributes>
</bm:transaction>
```

```
</bm:createTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

---

addToTransaction - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:addToTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:items>
<bm:catalogItem>
<bm:segment/>
<bm:product_line/>
<bm:model/>
</bm:catalogItem>
<bm:partItem>
<bm:part/>
<bm:quantity>1</bm:quantity>
<bm:price_book_var_name/>
</bm:partItem>
</bm:items>
<bm:transaction>
<bm:process_var_name>quickstart_commerce_process_bmClone_1</bm:process_var_name>
<bm:id/>
<bm:action_data>
<bm:document_var_name/>
</bm:action_data>
<bm:return_specific_attributes>
<bm:documents>
```

```
<bm:document>
<bm:var_name>quote_process</bm:var_name>
<bm:attributes>
<bm:attribute>_document_number</bm:attribute>
</bm:attributes>
</bm:document>
</bm:documents>
</bm:return_specific_attributes>
</bm:transaction>
<bm:include_transaction>true</bm:include_transaction>
</bm:addToTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

---

removeFromTransaction- Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</b-
m:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:removeFromTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:items>
<bm:item>
<bm:document_number/>
</bm:item>
</bm:items>
<bm:transaction>
<bm:process_var_name>quickstart_commerce_process_bmClone_1</bm:process_var_name>
<bm:id>6548624</bm:id>
<bm:action_data>
<bm:document_var_name/>
```

```
</bm:action_data>
</bm:transaction>
</bm:removeFromTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

---

updateTransaction - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:updateTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:transaction>
<bm:id>6548624</bm:id>
<bm:process_var_name>quickstart_commerce_process_bmClone_1</bm:process_var_name>
//additional fields not shown


</bm:transaction>
</bm:updateTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

---

getTransaction - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
```

```
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:transaction>
<bm:id/>
<bm:return_specific_attributes>
<bm:documents>
<bm:document>
<bm:var_name>quote_process</bm:var_name>
<bm:attributes>
<bm:attribute>_document_number</bm:attribute>
</bm:attributes>
</bm:document>
</bm:documents>
</bm:return_specific_attributes>
</bm:transaction>
</bm:getTransaction>
</soapenv:Body>
</soapenv:Envelope>
```

---

exportFileAttachment (content)- Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:exportFileAttachments xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<bm:mode>content</bm:mode>
<bm:inline/>
<bm:attachments>
<bm:attachment>
<bm:document_number/>
<bm:variable_name/>
</bm:attachment>
</bm:attachments>
<bm:transaction>
<bm:process_var_name>quickstart_commerce_process_bmClone_1</bm:process_var_name>
<bm:id/>
</bm:transaction>
</bm:exportFileAttachments>
</soapenv:Body>
</soapenv:Envelope>
```

---

exportFileAttachment (metadata) - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>08E2B1845C6FCDEB193274D29E923C7C</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://nqa1.bigmachines.com/bmfsweb/nqa1/schema/v1_0/commerce/transaction_bmClone_2.xsd</b-
m:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:exportFileAttachments xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:mode>metadata</bm:mode>
<bm:attachments>
<bm:attachment>
<bm:document_number>1</bm:document_number>
<bm:variable_name>fAtt</bm:variable_name>
</bm:attachment>
</bm:attachments>
<bm:transaction>
```

```
<bm:process_var_name>transaction_bmClone_2</bm:process_var_name>
<bm:id>12629013</bm:id>
</bm:transaction>
</bm:exportFileAttachments>
</soapenv:Body>
</soapenv:Envelope>
```

---

exportFileAttachemnt (base64)

Click here.

---

importFileAttachments (update) - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:importFileAttachments xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:mode>update</bm:mode>
<bm:attachments>
<bm:attachment>
<bm:document_number/>
<bm:variable_name/>
<bm:filename/>
<bm:file_content/>
</bm:attachment>
</bm:attachments>
<bm:transaction>
<bm:process_var_name>quickstart_commerce_process_bmClone_1</bm:process_var_name>
<bm:id/>
</bm:transaction>
</bm:importFileAttachments>
```

```
</soapenv:Body>
</soapenv:Envelope>
```

importFileAttachment (delete) - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/commerce/quickstart_commerce_process_bmClone_1.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:importFileAttachments xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:mode>delete</bm:mode>
<bm:attachments>
<bm:attachment>
<bm:document_number/>
<bm:variable_name/>
</bm:attachment>
</bm:attachments>
<bm:transaction>
<bm:process_var_name>quickstart_commerce_process_bmClone_1</bm:process_var_name>
<bm:id/>
</bm:transaction>
</bm:importFileAttachments>
</soapenv:Body>
</soapenv:Envelope>
```

importFileAttachment (base64)

Click here.



Those tags that are highlighted are required.

Line items must start with numerical sequence

What	Description
Scope	Blank line item attributes for every line item might only be necessary when Update Transaction is used with the _update_line_items action. For other actions, an empty <bm:sub_documents> tag is enough.
Cause	An Update Transaction SOAP call contains data for specific line items.
Solution 1	The document number is mandatory for every line item. Every line item's document number must be passed to the SOAP call, even for line items that do not require any updating. Simply pass these line items with the document number and blank attribute tags.
Solution 2	Make the SOAP call with a line-level action (as opposed to a Quote level action), which does not require all line items on the quote to be "considered" when parsing the input data. Note that this approach is not going to work if you need to update quote level attributes based on the updated line item attributes (e.g. pricing subtotals).

RELATED TOPICS

SOAP CONFIGURATION API

OVERVIEW

You can generate a skeleton input for each operation in the configuration WSDL, which contains 5 operations:

- configure
- getBOM
- getPrice
- getSpare
- deploy

For each operation, there are 3 fields that are always required in the header:

- sessionId
- category: Configuration
- schemaLocation

View API Details

API	Description	Required Input Variable	Success Response Variable	Fail Response Variable

configure	Takes input of a particular product family, line and model and generates a configuration. The response changes based on the options specified. For example, if SOAP XML mandates generating a transaction for the given configuration, then a transaction of the specified process is included in the response. Similarly, other details can be chosen to be included or not included in the response.	• sessionId
		• category
		• schemaLocation
		• segment
		• product_line
		• model
		• var_name attribute
		• price_book
		• var_name
			• status
			  • success
			  • message
			• item
			  • segment
			  • product_line
			  • model
				excetionCode
				exceptionMessage

getBom	Returns the BOM for the model.	• sessionId
		• category
		• schemaLocation
		• segment
		• product_line
		• model
			• status
			  • success
			  • message
			• item
			  • segment
			  • product_line
			  • model
			• attributes
				excetionCode
				exceptionMessage

API	Description	Required Input Variable	Success Response Variable	Fail Response Variable

			• flow
			• bom
			  • rule
			  • item
			  • partNumber
			  • quantity	

getPrice	Return the base price of the model.	• sessionId
		• category
		• schemaLocatio
		• segment
		• product_line
		• model
			• status
			  • success
			  • message
			• item
			  • segment
			  • product_line
			  • model
			• attributes
			• flow
			• price
			  • totalPrice
			  • attribute
			  • value
				excetionCode
				exceptionMessage

getSpare	Returns the recommended items for the model.	• sessionId
		• category
		• schemaLocation
		• segment
		• product_line
		• model
		• price_book_var_name
			• status
			  • success
			  • message
			• item
			  • segment
			  • product_line
			  • model
			• attributes
			• flow
			• spare
			  • rule
			  • comments
			  • item
			  • mandatory
			  • quantity
				excetionCode
				exceptionMessage

deploy	Deploys a specific product family.	• sessionId
		• category
		• schem-
			• status
			  • success
			  • message
				excetionCode
				exceptionMessage

API	Description	Required Input Variable	Success Response Variable	Fail Response Variable
		aLocation • product_family • var_name • attribute		

Returning Specific Attributes in Configuration APIs

BigMachines allows you to return specific attributes from a web service request with the <bm:return_specific_attributes /> tag. For example, you can pull in four main-document attributes and two sub-document attributes and a shortened version of the transaction XML, displaying only those attributes, will be returned.



This will only be allowed with the configure API.

ADMINISTRATION

Accessing Users WSDL and generating skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Configuration** tab.
3. Choose a configuration API from the **Select API** drop-down menu.
4. Select **one** of the following:
   - **Punchin URL** and enter the URL, or
   - **Product Family** and choose the product family, product line, and model from the corresponding drop-down menus.
5. Click **Generate Input** to retrieve the skeleton.

USE CASES

Example of Configuration APIs:

configure - Input SOAP XML

The following is the Minimum XML Input needed to configure a model only (will not create a commerce transaction or a quote, this XML also excludes the Price Book Tag):

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midSizedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:configure xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
</bm:item>
<bm:responseIncludes>
<bm:transaction>
<bm:process_var_name>standard_commerce_process</bm:process_var_name> <!--This refers to the variable name of the commerce process.-->
<bm:document_var_name>quote_line</bm:document_var_name> <!--This refers to the variable name of the commerce document. It's commonly referred to as either Quote (main) or Line (sub)-->
<bm:id>21185424</bm:id> <!--Transaction Number -->
</bm:transaction>
</bm:responseIncludes>
<bm:price_book_var_name>_default_price_book</bm:price_book_var_name> <!--This is included if you need to create a transaction or a quote if the price book is enabled. -->
<bm:attributes/>
</bm:configure>
</soapenv:Body>
</soapenv:Envelope>
```

---

getBOM - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header> <bm:userInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
```

```
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/config/hiTech/managedHosting_managedHosting.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getBom xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>hiTech</bm:segment>
<bm:product_line>managedHosting</bm:product_line>
<bm:model>managedHosting</bm:model>
</bm:item>
<bm:attributes/>
</bm:getBom>
</soapenv:Body>
</soapenv:Envelope>
```

---

getPrice - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midSizedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getPrice xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
</bm:item>
<bm:attributes/>
</bm:getPrice>
```

```
</soapenv:Body>
</soapenv:Envelope>
getBom - Input SOAP XML
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getBom xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
</bm:item>
<bm:attributes/>
</bm:getBom>
</soapenv:Body>
</soapenv:Envelope>
```

---

getSpare - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
```

```
<soapenv:Body>
<bm:getSpare xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item>
<bm:segment>sUVSuperstore</bm:segment>
<bm:product_line>midSizedSUV</bm:product_line>
<bm:model>midSizedSUV</bm:model>
<bm:mandatory />
<bm:quantity />
</bm:item>
<bm:price_book_var_name>_default_price_book</bm:price_book_var_name>
<bm:attributes/>
</bm:getSpare>
</soapenv:Body>
</soapenv:Envelope>
deploy - Input SOAP XML
Note: Deploy will is available for only
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>9ED35D4B82D521B3650084C851D86B42</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/config/sUVSuperstore/midSizedSUV_midS-
izedSUV.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:deploy xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:product_family>survey</bm:product_family>
<bm:emailTo/>
</bm:deploy>
</soapenv:Body>
</soapenv:Envelope>
```

---

deploy - Input SOAP XML

When you select deploy, you'll no longer have the option to select Punchin URL, you will just select the Product Family you wish to deploy.

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Configuration</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com"> <bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/config/hiTech/managedHosting_managedHosting.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getBom xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<bm:item> <bm:segment>hiTech</bm:segment>
<bm:product_line>managedHosting</bm:product_line>
<bm:model>managedHosting</bm:model>
</bm:item>
<bm:attributes/>
</bm:getBom>
</soapenv:Body>
</soapenv:Envelope>
```

RELATED TOPICS

SOAP DATA TABLES

OVERVIEW

You can generate a skeleton input for each operation in the data tables WSDL, which contains 5 operations:

- add
- update
- get
- delete
- deploy

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category: Data Table
- schemaLocation

View API Details

API	Description	Required Input Variable	Success Response Variable	Fail Response Variable

add	Adds a record to a data table.	• sessionId
		• category
		• schemaLocation
		• table_name
		Rows that need to be added (example):
		<bm:each_record>
		<bm:-currencyCode>USD</bm:CurrencyCode>
		<!--currencyCode is a column in the table-->
		<bm:-conversionRate>1.0</bm:converstionRate><!--coversionRate is a column in the table -->
		</bm:each_record>
			• status
			  • success
			  • message
				excetionCode
				excep-tionMessage

get	Returns a specified record in the criteria from the selected data table. You can only retrieve 300 rows at a time using a a 'get' call on a data table.	• sessionId
		• category
		• schemaLocation
		• table_name
			• status
			  • success
			  • message
			• Data Tables
				excetionCode
				excep-tionMessage

API	Description	Required Input Variable	Success Response Variable	Fail Response Variable
		• field • value • comparator	• each_)record • table_name	

update	Adds an existing record in a given data table. If you performing an update and your natural keys do not match, it will result in an error message.	• sessionId
		• category
		• schemaLocatio
		• table_name
		• rows that need to be updated
		  • sessionID
		  • category
			• status
			  • success
			  • message
				excetionCode
				exceptionMessage

delete	Deletes a record in a given data table.	• sessionId
		• category
		• schemaLocation
		• table_name
		• field
		• value
		• comparator
			• status
			  • success
			  • message
				excetionCode
				exceptionMessage

deploy	Deploys a given data table and adds an entry in the script table about the selected data table.	• sessionId
		• category
		• schemaLocation
		• table_name
			• status
			  • success
			  • message
				excetionCode
				exceptionMessage

Supported basic SQL standard comparators

Comparator	Notes
<	Less than
>	Greater than
=	Equals
!=	Not Equals. With CDATA tags, we can use <>
Like	Use for string fields.
In	Use for locating numeric data
Where Clause	

ADMINISTRATION

Accessing Users WSDL and generating the skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Data Tables** tab.
3. Choose a data table API from the **Select API** drop-down menu.
4. Choose a **Data Table** from the menu.
5. Enter a **Record ID**, if required.
6. Click **Generate Input** to retrieve the skeleton.

USE CASES

Example of Configuration APIs:

add- Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Data Tables</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/datatables/ApproverEmails.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:add xmlns:bm="urn:soap.bigmachines.com">
<bm:DataTables bm:table_name="ApproverEmails">
<bm:each_record>
<bm:approvalStep>1</bm:approvalStep>
<bm:emailAddress>jtam@bigmachines.com</bm:emailAddress>
<bm:is_valid>1</bm:is_valid>
</bm:each_record>
</bm:DataTables>
</bm:add>
</soapenv:Body>
</soapenv:Envelope>
```

---

update- Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Data Tables</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/datatables/ApproverEmails.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:update xmlns:bm="urn:soap.bigmachines.com">
<bm:DataTables bm:table_name="ApproverEmails">
<bm:each_record>
<bm:id>4133257</bm:id>
<bm:approvalStep>1</bm:approvalStep>
<bm:emailAddress>jtam@bigmachines.com</bm:emailAddress>
<bm:is_valid>1</bm:is_valid>
</bm:each_record>
</bm:DataTables>
</bm:update>
</soapenv:Body>
</soapenv:Envelope>
```

---

delete - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Data Tables</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/datatables/ApproverEmails.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:update xmlns:bm="urn:soap.bigmachines.com">
<bm:DataTables bm:table_name="ApproverEmails">
```

```
<bm:each_record>
<bm:id>4133257</bm:id>
<bm:approvalStep>1</bm:approvalStep>
<bm:emailAddress>jtam@bigmachines.com</bm:emailAddress>
<bm:is_valid>1</bm:is_valid>
</bm:each_record>
</bm:DataTables>
</bm:update>
</soapenv:Body>
</soapenv:Envelope>
```

---

get - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Data Tables</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/datatables/ApproverEmails.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:get xmlns:bm="urn:soap.bigmachines.com">
<bm:DataTables bm:table_name="ApproverEmails">
<bm:criteria>
<bm:field>approvalStep</bm:field>
<bm:value>1</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>emailAddress</bm:field>
<bm:value>jtam@bigmachines.com</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_added</bm:field>
<bm:value>2008-12-01 15:57:13</bm:value>
```

```
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_modified</bm:field>
<bm:value>2009-09-29 04:28:25.0</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
</bm:DataTables>
</bm:get>
</soapenv:Body>
</soapenv:Envelope>
```

deploy - Input SOAP XML

When you select deploy, you'll no longer have the option to select Punchin URL, you will just select the Product Family you wish to deploy.

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>12E05E5C9FE4F5EF0C1D07E006B465F1</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Data Tables</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/datatables/ApproverEmails.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:deploy xmlns:bm="urn:soap.bigmachines.com">
<bm:DataTables bm:table_name="ApproverEmails"/>
</bm:deploy>
</soapenv:Body>
</soapenv:Envelope>
```

RELATED TOPICS

SOAP EXCHANGE RATES API

OVERVIEW

Exchange rates are applied to currency attributes in both Configuration and Commerce based on the user's currency preference (or session / transaction currency preference). You can generate a skeleton input for each operation in the exchange rates WSDL, which contains 3 operations:

- modifyExchangeRate
- deleteExchangeRate
- getExchangeRate

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category: Exchange Rates
- schemaLocation

View API Details

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
modifyExchangeRate	Allows the user the ability to modify the conversion rate and uplift rate from the base currency to any other supported currency.	sessionID category schemaLocation sourceCurrencyCode targetCurrencyCode conversionRate Note: conversionRate is limited to 4 decimal points.	**status:** success message	exceptionCode exceptionMessage
deleteExchangeRate	Allows the user to delete a conversion rate and/or uplift rate for the specified currency. For example, if its no longer supported.	sessionID category schemaLocation targetCurrencyCode	**status:** success message	exceptionCode exceptionMessage
getExchangeRate	Allows the user to get the value of the conversion rate and uplift rate for the specified currency. This user can then determine if it needs to be updated.	sessionID category schemaLocation sourceCurrencyCode targetCurrencyCode	**status:** success message	exceptionCode exceptionMessage

ADMINISTRATION

Accessing Users WSDL and generating skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Exchange Rate** tab.
3. Choose from the **Select API** drop-down menu.
4. Click **Generate Input** to retrieve the skeleton.

USE CASES

Example of Users APIs:

modifyExchangeRate - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Exchange Rates</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/exchangerates/Exchange_Rates.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:modifyExchangeRate xmlns:bm="urn:soap.bigmachines.com">
<bm:exchangeRates>
<bm:each_record>
<bm:sourceCurrencyCode>USD</bm:sourceCurrencyCode>
<bm:targetCurrencyCode>AUD</bm:targetCurrencyCode>
<bm:conversionRate/>
<bm:upliftRate/>
</bm:each_record>
<bm:each_record>
<bm:sourceCurrencyCode>USD</bm:sourceCurrencyCode>
<bm:targetCurrencyCode>CAD</bm:targetCurrencyCode>
<bm:conversionRate/>
<bm:upliftRate/>
```

```
</bm:each_record>
<bm:each_record>
<bm:sourceCurrencyCode>USD</bm:sourceCurrencyCode>
<bm:targetCurrencyCode>CHF</bm:targetCurrencyCode>
<bm:conversionRate/>
<bm:upliftRate/>
</bm:each_record>
<bm:each_record>
<bm:sourceCurrencyCode>USD</bm:sourceCurrencyCode>
<bm:targetCurrencyCode>EUR</bm:targetCurrencyCode>
<bm:conversionRate/>
<bm:upliftRate/>
</bm:each_record>
<bm:each_record>
<bm:sourceCurrencyCode>USD</bm:sourceCurrencyCode>
<bm:targetCurrencyCode>GBP</bm:targetCurrencyCode>
<bm:conversionRate/>
<bm:upliftRate/>
</bm:each_record>
<bm:each_record>
<bm:sourceCurrencyCode>USD</bm:sourceCurrencyCode>
<bm:targetCurrencyCode>JPY</bm:targetCurrencyCode>
<bm:conversionRate/>
<bm:upliftRate/>
</bm:each_record>
</bm:exchangeRates>
</bm:modifyExchangeRate>
</soapenv:Body>
</soapenv:Envelope>
```

---

deleteExchangeRate - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Exchange Rates</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
```

```
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/exchangerates/Exchange_Rates.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:deleteExchangeRate xmlns:bm="urn:soap.bigmachines.com">
<bm:exchangeRates>
<bm:each_record>
<bm:targetCurrencyCode/>
</bm:each_record>
</bm:exchangeRates>
</bm:deleteExchangeRate>
</soapenv:Body>
</soapenv:Envelope>
```

---

getExchangeRate - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Exchange Rates</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/exchangerates/Exchange_Rates.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getExchangeRate xmlns:bm="urn:soap.bigmachines.com">
<bm:exchangeRate>
<bm:sourceCurrencyCode>USD</bm:sourceCurrencyCode>
<bm:targetCurrencyCode/>
</bm:exchangeRate>
</bm:getExchangeRate>
</soapenv:Body>
</soapenv:Envelope>
```



Those tags that are highlighted are required.

RELATED TOPICS

SOAP GROUPS API

OVERVIEW

The Groups SOAP API allows fullaccessusers to add or modify groups. Using web services enables you to manage user and group information on a large scale. User information and groups can be modified using two moethods. You can generate a skeleton input for each operation in the groups WSDL, which contains 2 operations:

- getGroups
- modifyGroups

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category:Groups
- schemaLocation

View API Details

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
getGroups	Allows you to obtain group information.	sessionID category schemaLocation variable_name	**status:** success message **groups** • group	exceptionCode exceptionMessage
modifyGroups	Allows you to create a new group and add users to the group. You can also add or remove user(s) to/from an existing group(s).	sessionID category schemaLocation variable_name	**status:** success message	exceptionCode exceptionMessage

---

ADMINISTRATION

Accessing Users WSDL and generating skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Groups** tab.
3. Choose from the **Select API** drop-down menu.
4. Click **Generate Input** to retrieve the skeleton.

USE CASES

getGroups - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Groups</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/groups/Groups.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getGroups xmlns:bm="urn:soap.bigmachines.com">
<bm:groups>
<bm:group>
<bm:company_name>Pm</bm:company_name>
<bm:variable_name/>
</bm:group>
</bm:groups>
</bm:getGroups>
</soapenv:Body>
</soapenv:Envelope>
```

---

modifyGroups - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Groups</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/groups/Groups.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:modifyGroups xmlns:bm="urn:soap.bigmachines.com">
```

```
<bm:groups>
<bm:group>
<bm:company_name>Pm</bm:company_name>
<bm:variable_name/>
<bm:name/>
<bm:users>
<bm:user>
<bm:action>add</bm:action>
<bm:login>user1</bm:login>
</bm:user>
<bm:user>
<bm:action>remove</bm:action>
<bm:login>user2</bm:login>
</bm:user>
</bm:users>
</bm:group>
</bm:groups>
</bm:modifyGroups>
</soapenv:Body>
</soapenv:Envelope>
```



Those tags that are highlighted are required.

RELATED TOPICS

SOAP PRICE BOOK ASSOCIATION API

OVERVIEW

You can generate a skeleton input for each operation of the price book associations WSDL, which contains 5 operations:

- add
- update
- get
- delete
- deleteAll

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category: Security
- schemaLocation

View API Details

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
add	Supports adding price book associations; assuming both the price books and parts exist in the system.	sessionID category schemaLocation price_book_var_name part_number	**status:** success message	exceptionCode exceptionMessage
get	Supports getting specific associations based on desired criteria. Note: multiple criterion will be joined with an 'AND' in the query, so try to specify a single criterion. A part number is required.	sessionID category schemaLocation price_book_var_name field value comparator	**status:** success message **PriceBookAssocs:** each_record • part_number • price_book_var_entry_id	exceptionCode exceptionMessage
update	Updates an association between a given price book	sessionID	**status:**	exceptionCode

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
	and a part. A part number is required.	category schemaLocation price_book_var_name part_number	success message	exceptionMessage
delete	Supports the deletion of associations based on desired criteria. Note: multiple criterion will be joined with an 'AND' in the query, so try to specify a single criterion. A part number is required.	sessionID category schemaLocation price_book_var_name field value comparator	**status:** success message	exceptionCode exceptionMessage
deleteAll	Supports deleting ALL price book associations for a price book.	sessionID category schemaLocation price_book_var_name	**status:** success message	exceptionCode exceptionMessage

ADMINISTRATION

Accessing Users WSDL and generating skeleton input

1. Navigate from: Admin Home Page > Integration Platform > Web Services.
2. Click on the **Price Book Assocs** tab.
3. Choose from the **Select API** drop-down menu.
4. Choose a **Price Book** from the drop-down menu.
5. Click **Generate Input** to retrieve the skeleton.

USE CASES

Example of Users APIs:

add - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
```

```
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Price Book Assocs</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/pricebooks/PriceBookAssocs.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:add xmlns:bm="urn:soap.bigmachines.com">
<bm:PriceBookAssocs bm:price_book_var_name="westCoastPricing"/>
</bm:add>
</soapenv:Body>
</soapenv:Envelope>
```

---

get - Input SOAP XML

You must enter a Part Number before clicking Generate Input

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Price Book Assocs</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/pricebooks/PriceBookAssocs.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:get xmlns:bm="urn:soap.bigmachines.com">
<bm:PriceBookAssocs bm:price_book_var_name="westCoastPricing">
<bm:criteria>
<bm:field>part_number</bm:field>
<bm:value>RAM-128</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_added</bm:field>
<bm:value>2010-11-23 11:06:24</bm:value>
```

```
<bm:comparator>=</bm:comparator>
</bm:criteria>
<bm:criteria>
<bm:field>date_modified</bm:field>
<bm:value>2012-03-24 05:11:45</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
</bm:PriceBookAssocs>
</bm:get>
</soapenv:Body>
</soapenv:Envelope>
```

---

update - Input SOAP XML

You must enter a Part Number before clicking Generate Input

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Price Book Assocs</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/pricebooks/PriceBookAssocs.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:update xmlns:bm="urn:soap.bigmachines.com">
<bm:PriceBookAssocs bm:price_book_var_name="westCoastPricing"/>
</bm:update>
</soapenv:Body>
</soapenv:Envelope>
```

---

delete - Input SOAP XML

You must enter a Part Number before clicking Generate Input

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
```

```
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Price Book Assocs</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/pricebooks/PriceBookAssocs.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:delete xmlns:bm="urn:soap.bigmachines.com">
<bm:PriceBookAssocs bm:price_book_var_name="westCoastPricing">
<bm:criteria>
<bm:field>part_number</bm:field>
<bm:value>RAM-128</bm:value>
<bm:comparator>=</bm:comparator>
</bm:criteria>
</bm:PriceBookAssocs>
</bm:delete>
</soapenv:Body>
</soapenv:Envelope>
```

---

deleteAll - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>A99EA9EA9FFC03C02AF30373A60CFD60</bm:sessionId>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Price Book Assocs</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://pm.bigmachines.com/bmfsweb/pm/schema/v1_0/pricebooks/PriceBookAssocs.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:deleteAll xmlns:bm="urn:soap.bigmachines.com">
<bm:PriceBookAssocs bm:price_book_var_name="westCoastPricing"/>
</bm:deleteAll>
</soapenv:Body>
</soapenv:Envelope>
```



Those tags that are highlighted are required.

RELATED TOPICS

SOAP SECURITY API

OVERVIEW

You can generate a skeleton input for each operation in the security WSDL, which contains 4 operations:

- login
- getUserInfo
- setSessionCurrency
- logout

For each operation, there are 3 fields that are always required in the header:

- sessionID
- category: Security
- schemaLocation

View API Details

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
login	Adds a user through SOAP. When multiple users are created using this method, the entire request is processed before a response is sent. Allows a session ID to be created for a user based on the username and password. This operation expects input of username and password as parameters and returns the sessionId upon successful login, otherwise it returns an error message.	category schemaLocation username password	**status:** success message **sessionID**	exceptionCode exceptionMessage
getUserInfo	Does not define any request parameter in the SOAP body, but operates on the information of sessionID, specified in the header.	sessionID category schemaLocation	**status:** success message **userInfo:** login first_name group_list • group • variable_name	exceptionCode exceptionMessage
setSessionCurrency	Does not define any request parameter in the SOAP body, but operates on the information of sessionID, specified in the header.	sessionID category	**status:** success	exceptionCode exceptionMessage

API	Description	Required Input Variables	Success Response Variables	Failure Response Variables
		schemaLocation sessionCurrency	message	
logout	Allows the user to logout of BigMachines using SOAP.	sessionID category schemaLocation	**status:** success message	exceptionCode exceptionMessage

ADMINISTRATION

Accessing Users WSDL and generating skeleton input

1. Navigate from Admin Home Page > Integration Platform > Web Services.
2. Click on the **Security** tab.
3. Choose from the **Select API** drop-down menu.
4. Click **Generate Input** to retrieve the skeleton.

USE CASES

Example of Users APIs:

login - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Security</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>myCompany.bigmachines.com/bmfsweb/myCompany/schema/v1_0/security/Security.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:login xmlns:bm="urn:soap.bigmachines.com">
<bm:userInfo>
<bm:username>%username%</bm:username> //The username of a valid, active user within BigMachines
<bm:password>%password%</bm:password> //The password associated with the above username
<bm:sessionCurrency />
```

```
</bm:userInfo>
</bm:login>
</soapenv:Body>
</soapenv:Envelope>
```

---

getUserInfo - Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>%sessionId%</bm:sessionid>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Security</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/security/Security.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:getUserInfo xmlns:bm="urn:soap.bigmachines.com"/>
</soapenv:Body>
</soapenv:Envelope>
```

---

setSessionCurrency- Input SOAP XML

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionId>%sessionID%</bm:sessionID>
</bm:userInfo>
<bm:category xmlns:bm="urn:soap.bigmachines.com">Security</bm:category>
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">
<bm:schemaLocation>https://myCompany.bigmachines.com/bmfsweb/pm/schema/v1_0/security/Security.xsd</bm:schemaLocation>
</bm:xsdInfo>
</soapenv:Header>
<soapenv:Body>
<bm:setSessionCurrency xmlns:bm="urn:soap.bigmachines.com">
<bm:sessionCurrency/>
</bm:setSessionCurrency>
```

```
</soapenv:Body>

</soapenv:Envelope>
```

logout - Input SOAP XML

```
<xsi="www.w3.org/2001/XMLSchema-instance" xsd="www.w3.org/2001/XMLSchema" soapenv="schemas.xmlsoap.org/soap/envelope/" encoding="UTF-8">

<soapenv:Header>

<bm:userInfo xmlns:bm="urn:soap.bigmachines.com" soapenv:actor="schemas.xmlsoap.org/soap/actor/next" soapenv:mustUnderstand="0">

<bm:sessionId>%sessionID%</bm:sessionID>

</bm:userInfo>

<bm:category xmlns:bm="urn:soap.bigmachines.com" soapenv:actor="schemas.xmlsoap.org/soap/actor/next" soapenv:mustUnderstand="0">Security</bm:category>

<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com" soapenv:actor="schemas.xmlsoap.org/soap/actor/next" soapenv:mustUnderstand="0">

<bm:schemaLocation>njupiter.bigmachines.com/bmfsweb/njupiter/schema/v1_0/security/Security.xsd</bm:schemaLocation>

</bm:xsdInfo>

</soapenv:Header>

<soapenv:Body>

<bm:logout xmlns:bm="urn:soap.bigmachines.com"/>

</soapenv:Body>

</soapenv:Envelope>
```



Those tags that are highlighted are required.

RELATED TOPICS

PREPARE A SOAP REQUEST

OVERVIEW

We need to build out the SOAP request. We recommend using a tool such as SoapUI for this. SoapUI takes in the WSDL as input and generates the SOAP request for each API call defined by the WSDL. All we have to do, is take this SOAP request, change it to have the fields we require, and send it through a POST HTTP request.

ADMINISTRATION

Let's assume we have the required WSDL files containing the definitions of the various web services available. Secondly, let us assume we have the session ID.

1. Begin with the following SAMPLE input SOAP XML.

```
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/' xmlns:n-
s='urn:crmondemand/ws/account/10/2004' xmlns:acc='urn:/crmondemand/xml/account'>
<soapenv:Body>
<ns:AccountWS_AccountQueryPage_Input>
<ns:PageSize>100</ns:PageSize>
<acc:ListOfAccount>
<acc:Account>
<acc:AccountId/>
<acc:Description/>
<acc:Location/>
<acc:AccountName>LIKE '*BAKE*'</acc:AccountName>
<acc:Region/>
<acc:AccountType/>
<acc:ListOfAddress>
<acc:Address>
<acc:AddressId/>
<acc:City/>
<acc:Country/>
<acc:StateProvince/>
<acc:Address/>
<acc:StreetAddress2/>
<acc:StreetAddress3/>
</acc:Address>
</acc:ListOfAddress>
</acc:Account>
</acc:ListOfAccount>
</ns:AccountWS_AccountQueryPage_Input>
</soapenv:Body>
</soapenv:Envelope>
```

2. Put this SOAP input into a POST request. In BigMachines we would do this through URLDATABYPOST.
3. "Stringify" the above SOAP input, and that forms second parameter to URLDATABYPOST.
4. The first parameter is the Url. That would also be available from the WSDLs and along with the session id, should look something like:

```
https://secure-ausomxdia.crmondemand.com/Services/Integration/object;jsession=8d497390b14d3588fef9b6f.e3iRbuMa
```

RELATED LINKS

DEBUGGING SOAP RESPONSES

OVERVIEW

This article will give you some tips for debugging SOAP responses.

ADMINISTRATION

1. Check the Error logs. Here's what we are most likely to see:

   …Could open connection.java.io.IOException: Server returned HTTP response code: 400 for URL: https://secure-aus-omxdia.crmondemand...on/object;jses...

2. Let's review/verify the following:
   - We used a valid session
   - We built the SOAP input
   - We prepared the URL
   - We made a POST HTTP request.
3. Use Firebug or Fiddler to debug the request. *In this case, we're using Fiddler.*
4. Build out the entire request to see what the 400 actually refers to.
5. Go to the Request Builder in Fiddle and add the required URL along with the session ID.
6. Change the request type to POST.
7. Pass the soapInput in the post data.
8. Execute the request. *The first thing you may notice is that headers are required.*
9. Add the content type header and try again. This time, look at the response XML and notice that you need to add the SOAPAction.
10. Look at SoapUI and notice that it's header contains the following:
11. SOAPAction: "document/urn:crmondemand/ws/account/10/2004:AccountQueryPage"
12. Navigate back to BigMachines.
13. Add a fourth parameter to URLDATABYPOST, which will be a diction that can contain custom headers.
14. Add SOAPAction and Content-type to the parameter you just created. The script should resemble:

```
headerDict=dict('string');
```

```
put(headDict, "Content-Type", "text/xml:charset=utf-8");
put(headDict, "SOAPAction",
"\"document/urn:crmondemand/ws/account/10/2004:AccountQueryPage\"");
```

NOTES



Don't forget to escape the double quotes (\") in the SOAPAction header.

RELATED TOPICS

CALLS TO ORACLE ON DEMAND FROM CONFIGURATION

ADMINISTRATION

Setting up Calls to Oracle On Demand

1. Prepare a session by locating _BM_USER_PARTNER_SESSION_ID on doing a partner login.
2. Preparing the SOAP Request. *Let's assume we have the required WSDL files containing the definitions of the various web services available. Secondly, let us assume we have the session Id.*
3. Build the SOAP request.
4. Use a tool such as SoapUI for this. SoapUI takes in the WSDL as input and generates the SOAP request for each API call defined by the WSDL. All we have to do, is take this SOAP request, change it to have the fields we require, and send it through a POST HTTP request. Let us break this down.
5. See SOAP input XML.
6. Put this SOAP input into a POST request.
7. In BigMachines we would do this through URLDATABYPOST. "Stringify" the above SOAP input, and that forms second parameter to URLDATABYPOST.
8. The first parameter is the Url. That would also be available from the WSDLs and along with the session id, should look something like:

   https://secure-ausomxdia.crmondemand.com/Services/Integration/object;jsession=8d497390b14d3588fef9b6f.e3iRbuMa
9. Using URLDATABYPOST we should have;

   ```
   responseXmlString = urldatabypost(url, soapInput, "Default Response on error");
   ```
10. BigMachines makes a POST HTTP request passing the soapInput in the body of the request. On getting a response, we notice the default error message. OOPS! What just happened here? Thought we had everything in place. We should have got a nice little XML containing the data we requested for. SoapUI gives us exactly what we need, then why not BigMachines? What are we missing?

TROUBLESHOOTING

Debugging Errors

1. Navigate to the Error logs. Here's what we are most likely to see;

   …Could open connection.java.io.IOException: Server returned HTTP response code: 400 for URL: https://secure-aus-omxdia.crmondemand...on/object;jses...
2. Debug using Firebug or Fiddler. I just like the way Fiddler works, so lets go with that. Let us build out the entire request using Fiddler, and see what the 400 actually refers to.

a. Go to the Request Builder in Fiddler, add the required Url along with the session id, as explained earlier.

b. Change the request type to a POST.

c. In the post data pass the soapInput.

d. Execute the request.

3. First thing we notice, headers are required, we add the content type header and try again.

4. When the response comes in, as expected, again we see a 400 error. This time however, we can take a look at the response XML. On investigating we find out that another piece of information is missing in the header – “SOAPAction”.

5. Look back at SoapUI, and we notice that it’s header contains:

   SOAPAction: "document/urn:crmondemand/ws/account/10/2004:AccountQueryPage"

6. Now we come back to BigMachines, and thankfully we have a fourth parameter to URLDATABYPOST, a dictionary that can contain custom headers. We add SOAPAction and Content-type to it. The code should look something like:

```
headerDict = dict("string");

put(headerDict, "Content-Type", "text/xml; charset=utf-8");

put(headerDict, "SOAPAction", "\"document/urn:crmondemand/ws/account/10/2004:AccountQueryPage\"");
```



Don't forget to escape the double quotes (\") in the SOAPAction header.

RELATED TOPICS

CREATE QUOTES AND ASSIGN OWNERS

OVERVIEW

In this scenario, an external system must have the ability to create quotes and assign them an owner.

ADMINISTRATION

Creating quotes and assigning owners can be accomplished using 3 Web Service SOAP Calls:

1. login
2. createTransaction
3. updateTransaction



Typically an email is sent on the step transition to inform the user that a new quote has been created for them.

USE CASES

Click for an example of how the code should appear.

RELATED TOPICS

CREATE QUOTES AND ASSIGN OWNERS

OVERVIEW

In this scenario, an external system must have the ability to create quotes and assign them an owner.

ADMINISTRATION

Creating quotes and assigning owners can be accomplished using 3 Web Service SOAP Calls:

1. login
2. createTransaction
3. updateTransaction



Typically an email is sent on the step transition to inform the user that a new quote has been created for them.

USE CASES

Click for an example of how the code should appear.

RELATED TOPICS

DATA TABLE UPDATES

OVERVIEW

In this scenario, data tables require regular updates.

ADMINISTRATION

Data Table updates can be accomplished in 2 ways:

1. FTP Server Automation
    - Can delete entire table or just add new rows.
    - Recommended as a monthly, weekly, or daily feed.
2. Data Table Web Services
    - Get, delete, and/or update calls.
    - Recommended for very infrequent or event-based updates.

NOTES



FTP Server Automation is exponentially faster than Web Services.

RELATED TOPICS

OBTAIN CUSTOMER INFORMATION

OVERVIEW

In this scenario, an external system must have the ability to access customer information.

ADMINISTRATION

The following examples list several possible scenarios and details how to properly address each scenario.

1. Quotes are created using their current CPQ system and the information must be passed utilizing:
    - login
    - updateTransaction
    - createTransaction
2. Customer information is imported utilizing:
    - Real-time Integration
    - updateTransaction
3. BMI can integrate with tools to obtain customer information. The BigMachines user should be able to click a button to obtain the needed information in real-time, utilizing:
    - geturldatabypost

RELATED TOPICS

QUOTE INFORMATION

OVERVIEW

In this scenario, an external system must have the ability to access BigMachines quote information.

ADMINISTRATION

Accessing quote information via an external system can be accomplished in 2 ways:

1. Real-time Integration
    - Recommended approach, especially if a lot of data is needed.
1. getTransaction Web Service Call
    - Send transaction ID and a few parameters. Have their system call geTransaction.
    - Recommended if a web service already exists or very little information is needed. Send data in name value pairs.
    - XML

RELATED TOPICS

UPDATE PRICES

OVERVIEW

In this scenario, an external system must have the ability to update pricing.

ADMINISTRATION

The following examples list several possible scenarios and details how to properly address each scenario.

1. Prices update on a daily basis. Prices are contained in data tables with ~500,000 rows.
    - Use: FTP Server Automation
2. The wireless offering is updated on a weekly basis. The offering is stored in data tables with ~10,000 rows.
    - Use: FTP Server Automation
3. Orders must be sent downstream from BMI to an external ERP system.
    - Use: Real-time Integration
4. Prices for a given customer should be saved for future use.
    - Use: Access Self-Soap using "get" function. Delete and update Data Table Web Service calls.

RELATED TOPICS

UPDATE QUOTES

OVERVIEW

In this scenario, an external system must have the ability to update a given quote.

ADMINISTRATION

Updating a quote via an external system can be accomplished using 2 Web Service SOAP Calls:

1. Login
2. updateTransaction (which requires a transaction ID)

RELATED TOPICS

USE BML TO CONNECT TO AN EXTERNAL SERVICE

OVERVIEW

There are a number of external systems that BigMachines may be communicating with, including ERP Systems, CRM Systems, external databases, cgi scripts, and more. When use cases call for representing objects and complicated product or pricing systems, we can often meet the needs directly in the BigMachines application. However, when integrating with an ERP system or CRM, complexity may arise. Re-implementing existing services can be costly and cumbersome. And some domain-specific tasks are best dealt with in specific and targeted systems.

To deal with these needs, you can setup communication points between BigMachines and the external system in question. By default, we provide Integration points through an XSL interface to some CRM systems. When the pre-built mapping and integration tools don't meet your needs, then BML can be used to send and receive messages directly from a given service.

ADMINISTRATION

Before beginning

Always abstract your data calls into util functions. This simplifies the process of communicating with the external service, and makes it easier to change in the future if necessary. You can use either POST or GET to communicate with the external server, using the standard BML functions urldatabypost and urldatabyget.

There are a few things to consider when choosing which protocol to use:

- HTTP specifies that POST is used to make changes on a server.
- When avoidable, GET should not be used to make changes.
- GET is the same thing as putting the url directly into your browser, with the parameters appended to the end of the URL.
- GET has a max character limit based on URL length - generally 2048 characters.
- POST sends the data behind the scenes.
- POST doesn't have a set max character limit, and is good for large data transfers.
- Many services treat GET and POST identically.

Your choice will depend on what your needs are, as well as the specifications of the web service that you are communicating with.

ADMINISTRATION

Example: Google Movie API

In order to find out what movies are playing in our area, we're going to harness Google's Movie API. I know that the API calls for "GET," but still use makeurlparam whenever possible to make it easy to switch between GET and POST if necessary:

```
/**
* Util function Movies
* takes: location {String} - A Zip Code or City
* error {String} - Error message to return
```

```
*/
url = "http://www.google.com/ig/api";
param = makeurlparam({'movies':location});
result = urldatabyget(url, param, error);
return result;
```

Call the util function "Movies" from anywhere in my application, to find out what movies are playing in a given area.

TROUBLESHOOTING

Visit the Link Directly

1. If you are using GET, type the URL into your browser to see the response directly. You can change the parameters by playing with the link.
2. If you are using POST, one simple way to see the results is to create an HTML form and submit it.
3. Our earlier example could be turned into a form like so:

```
<!--test_submit.html-->
<form action="http://www.google.com/ig/api"method="post">
<input name="movies" value="San Francisco"/>
<input type="submit">
</form>
```

4. Save the form and open it in your browser, then hit the "Submit" button to send the post.
5. The results of the post will be loaded on the screen, and the Firebug "Net" panel will have additional information about the result.

Check your Parameters

You can pass a parameter object created by makeurlparams directly to the BML print statement, to see exactly what is being sent.

Check the Error Logs

Sometimes the BigMachines error logs will show information that the debugger doesn't.

Check for Firewalls and Security Settings

Sometimes you will be able to visit a link directly, but a security setting could be blocking the BigMachines server. Talk to the administrator of the web service to find out if there are any access restrictions in place. The Ops team may be able to help verify if the connection is occurring as well.

Special Content Types

Sometimes a web service demands a specific content type be set for a request to succeed. For instance, some SOAP calls may not work if the Content-Type header has not been explicitly set to "text/xml." If you are able to invoke the service manually, but your BML code is not able to connect, this could be the problem.

To correct this, you can pass a fourth parameter into the "urldatabyget" and "urldatabypost" functions in BML. This optional fourth parameter is a dictionary that can be used to set values on the HTTP header. **To set the header for SOAP content, write:**

```
headerDict = dict("string");

put(headerDict, "Content-Type", "text/xml; charset=utf-8");

soapResponse = urldatabypost(soapURL, params, error, headerDict);
```

The requirements for the header may vary based on the web service. If you can contact the administrator of the external service they may be able to provide this information; otherwise the Firebug "Net" panel includes header information that may assist in determining the requirements.

RELATED TOPICS

INTERNATIONALIZATION OVERVIEW

OVERVIEW

Internationalization (I18N) ensures that the system is able to provide support for multiple currencies, languages and numerical/date/time formats.The following features make BigMachines an internationalized software:

- Processing data in that language and storing it.
- Displaying different characters.
- Multi-byte character support.
- Support a variety of locales: currencies, date/time formats, numerical formats.

Unlimited Currency Support

BigMachines has removed the limit of concurrent currencies that could be supported per application. Previously limited to 24, users can now enable any and all of the currencies that BigMachines currently supports (See article on Multi-Currency Support for full list of supported currencies).

Translations Bundle

Working with a professional translation company, BigMachines has translated site content into the following languages:

- French
- German
- Spanish

All product templates and pre-existing BigMachines content users receive at implementation is available in translation.



The addition of the translations bundle does not affect current customer functionality.

User is not required to make any selections, once site language preference established, customer site will appear with translations in place.

NOTES



Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value*

will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.

If a translation does not exist, then the user is shown the data in the language that it was created in. All data and labels can be translated.

At least one fullaccessuser must have the same language preference as the base language of the site. Only this user is allowed to provides translations of all other supported languages.

Variable Names, Part Numbers and Customer ID do not support Multi-byte characters.

Default value translation occurs during the creation of the product line/model. This is what translations for name, text, text area and description fields of a product line or model are not observed by a user with different language preferences.

- For example, if a Chinese user creates and saves a product line, then he can see the description for the product line in Chinese. Now, when an English user logs in, and views this Product line on either the admin or the buy side, the value in the description fields will still be in Chinese. When the second user creates a second product line, then the description field will be in English. If the Chinese user tries to view the second product line, he will see the description in English.

RELATED TOPICS

INTERNATIONALIZATION OVERVIEW

OVERVIEW

Internationalization (I18N) ensures that the system is able to provide support for multiple currencies, languages and numerical/date/time formats.The following features make BigMachines an internationalized software:

- Processing data in that language and storing it.
- Displaying different characters.
- Multi-byte character support.
- Support a variety of locales: currencies, date/time formats, numerical formats.

Unlimited Currency Support

BigMachines has removed the limit of concurrent currencies that could be supported per application. Previously limited to 24, users can now enable any and all of the currencies that BigMachines currently supports (See article on Multi-Currency Support for full list of supported currencies).

Translations Bundle

Working with a professional translation company, BigMachines has translated site content into the following languages:

- French
- German
- Spanish

All product templates and pre-existing BigMachines content users receive at implementation is available in translation.



The addition of the translations bundle does not affect current customer functionality.

User is not required to make any selections, once site language preference established, customer site will appear with translations in place.

NOTES



Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value*

will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.

If a translation does not exist, then the user is shown the data in the language that it was created in. All data and labels can be translated.

At least one fullaccessuser must have the same language preference as the base language of the site. Only this user is allowed to provides translations of all other supported languages.

Variable Names, Part Numbers and Customer ID do not support Multi-byte characters.

Default value translation occurs during the creation of the product line/model. This is what translations for name, text, text area and description fields of a product line or model are not observed by a user with different language preferences.

- For example, if a Chinese user creates and saves a product line, then he can see the description for the product line in Chinese. Now, when an English user logs in, and views this Product line on either the admin or the buy side, the value in the description fields will still be in Chinese. When the second user creates a second product line, then the description field will be in English. If the Chinese user tries to view the second product line, he will see the description in English.

RELATED TOPICS

MULTI-CURRENCY SUPPORT

OVERVIEW

An application must have a base currency that is defined during set-up. Once a currency set is established, additional currencies can be added, but existing currencies cannot be removed. User accounts and external company accounts can designate a currency of their preference from the currency set supported by the application. The base currency is the default currency for all users. In order to enable or disable multiple currencies, please contact BigMachines OPS.

- Base currencies will be used to calculate other currencies in the conversion rates area. For pricing rules, or any time you are returning a price in a function, the conversion rates will run. The system will assume you are returning currency fields in *some currency* and it will run the conversion rates for transactions and users with other currencies as their preferred currency.
- Model and parts pricing can be entered into the system in all supported currencies.
- Exchange rate rules can be specified for currency conversion. Currency conversions do not trigger if there is a price defined in the currency preference of the buyer. Currency conversions happen in real-time when a price does not exist in the user's currency preference, but a base currency price exists along with an exchange rate to the user's currency preference.
- Commerce transactions use the currency preference of the document originator. Once set, a commerce transaction's currency cannot be changed. This ensures that all commerce documents within the same transaction have the same currency. Both buyers and suppliers can view and modify commerce documents that are in a different currency than their own currency preference.

List of Supported Currencies as of BigMachines 2013 Spring

Currency Code	Country	Currency
AED	United Arab Emirates	Emerati Dirham
AFA	Afghanistan	Afghani (deprecated)
AFN	Afghanistan	Afghani
ALL	Albania	Lek
AMD	Armenia	Armenian Dram
ANG	Curacao	Antillean Guilder
AOA	Angola	Kwanza
ARS	Argentina	Argentine Peso
ATS		Schilling (deprecated)
AUD	Australia	Australian Dollar
AWG	Aruba	Florin
AZN	Azerbaijan	Manat
BAM	Bosnia and Herzegovina	Convertible Mark
BBD	Barbados	Barbados Dollar
BDT	Bangladesh	Taka

Currency Code	Country	Currency
BEF		Franc (deprecated)
BGL		Leva (deprecated)
BGN	Bulgaria	Lev
BHD	Bahrain	Bahraini Dollar
BIF	Burundi	Burundi Franc
BMD	Bermuda	Bermudian Dollar
BND	Brunei	Brunei Ringgit
BOB	Bolivia	Bolivano
BOV	Bolivia	Bolivian Mvdol
BRL	Brazil	Brazilian Real
BSD	Bahamas	Bahamian Dollar
BTN	Bhutan	Ngultrum
BWP	Botswana	Pula
BYR	Belarus	Belarusian Ruble
BZD	Belize	Belize Dollar
CAD	Canada	Canadian Dollar
CDF	Congo, The Democratic Republic of	Congolese Franc
CFAF	Congo, The Democratic Republic of	Franc
CHF	Switzerland	Swiss Franc
CLF	Chile	Unidad de Fomento
CLP	Chile	Chilean Peso
CNY	China	Chinese Yan Reminbi
COP	Columbia	Columbian Peso
CRC	Costa Rica	Costa Rican Colon
CUP	Cuba	Cuban Peso
CVE	Cape Verde	Cape Verde Escudo
CZK	Czech Republic	Crown
DEM	Germany	Deutschmark (deprecated)
DJF	Djibouti	Dijiboutian Franc
DKK	Denmark	Krone
DOP	Dominican Republic	Dominican Peso
DZD	Algeria	Algerian Dinar

Currency Code	Country	Currency
EEK		Kroon (deprecated)
EGP	Egypt	Egyptian Pound
ERN	Eritrea	Nafka
ESP	Spain	Peseta (deprecated)
ETB	Ethiopia	Birr
EUR	European Union	Euro
FIM		Markka (deprecated)
FJD	Fiji	Fiji Dollar
FKP	Falkan Island	Falkand Islands Pound
FRF	France	Franc (deprecated)
GBP	United Kingdom	United Kingdom Pound
GEL	Georgia	Lari
GHS	Ghana	Cedi
GIP	Gibraltar	Gibraltar Pound
GMD	Gambia	Dalasi
GNF	Guinea	Guinean Franc
GRD		Drachma (deprecated)
GTQ	Guatamela	Quetzal
GYD	Guyana	Guyanese Dollar
HKD	Hong Kong	Hong Kong Dollar
HNL	Honduras	Lempira
HRK	Croatia	Kuna
HTG	Haiti	Gourde
HUF	Hungary	Forint
IDR	Indonesia	Indonesia Rupiah
IEP		Punt (deprecated)
ILS	Israel	New Shekel
INR	India	Indian Rupee
IQD	Iraq	Iraqian Dinar
IRR	Iran, Islamic Republic of	Rial
ISK	Iceland	Krona
ITL	Italy	Lira (deprecated)

Currency Code	Country	Currency
JMD	Jamaica	Jamaican Dollar
JOD	Jordan	Jordan Dinar
JPY	Japan	Japanese Yen
KES	Kenya	Kenya Shilling
KGS	Kyrgystan	Som
KHR	Cambodia	New Riel
KMF	Comoros	Comoro Franc
KPW	Korea, Democratic People's Republic of	North Korean Won
KRW	Korea, Republic of	South Korean Won
KWD	Kuwait	Kuwaiti Dinar
KYD	Cayman Islands	Cayman Islands Dollar
KZT	Kazakhstan	Tenge
LAK	Lao People's Democratic Republic	New Kip
LBP	Lebanon	Lebanese Pound
LKR	Sri Lanka	Sri Lanka Rupee
LRD	Liberia	Liberian Dollar
LSL	Lesotho	Loti
LTL	Lithunia	Litas
LUF	Luxemberg	Luxemberg Franc (deprecated)
LVL	Latvia	Lat
LYD	Libya	Libyan Dollar
MAD	Morocco	Dirham
MDL	Moldova, Republic of	Leu
MGA	Madagascar	Ariary
MGF	Madagascar	Malagasy Francs (deprecated)
MKD	Marcedonia	Denar
MLF	Mali	Mali Franc (deprecated)
MMK	Myannmar	Kyat
MNT	Mongolia	Tugrik
MOP	Macao	Pataca
MRO	Mauritania	Mauritius Rupee
MTL	Malta	Malta Lira (deprecated)

Currency Code	Country	Currency
MUR	Mauritius	Mauritius Rupee
MVR	Maldives	Rufiyaa
MWK	Malawi	Kwacha
MXN	Mexico	Peso
MXV	Mexico	Mexican Unidad de Inversion
MYR	Malaysia	Malaysian Ringgit
MZM	Mozambique	Metical (deprecated)
MZN	Mozambique	Metical
NAD	Namibia	Namibian Dollar
NGN	Nigeria	Naira
NIO	Nicaragua	Cordoba
NLG		Guilder (deprecated)
NOK	Norway	Norway Krone
NPR	Nepal	Nepalese Rupee
NZD	New Zealand	New Zealand Dollar
OMR	Oman	Omani Rial
PAB	Panama	Balboa
PEN	Peru	New Sol
PGK	Papau New Guinea	Kina
PHP	Philippines	Philippine Peso
PKR	Pakistan	Pakistani Rupee
PLN	Poland	Zloty
PLZ	Poland	Zloty (deprecated)
PTE		Escudo (deprecated)
PYG	Paraguay	Guarani
QAR	Qatar	Qatar Riyal
ROL		Leu (deprecated)
RON	Romania	Romania New Leu
RSD	Serbia	Serbian Dinar
RUB	Russia	Ruble
RWF	Rwanda	Rwanda Franc
SAR	Saudi Arabia	Riyal

Currency Code	Country	Currency
SBD	Solomon Islands	Solomon Island Dollar
SCR	Seychelles	Seychelles Rupee
SDP	Sudan	Sudanese Dinar (deprecated)
SEK	Sweden	Krona
SGD	Singapore	Singapore Dollar
SHP	Saint Helena	Saint Helena Pound
SIT		Tolar (deprecated)
SKK		Slovakia Koruna (deprecated)
SLL	Sierra Leone	Leone
SOS	Somalia	Somali Shilling
SRD	Suriname	Surinamese Dollar
STD	Sao Tome and Principe	Dobra
SYP	Syrian Arab Republic	Syrian Pound
SZL	Swaziland	Lilangeni
THB	Thailand	Baht
TJS	Tajikstan	Somoni
TMT	Turkmenistan	Turkmenistani Manat
TND	Tunisia	Tunisian Dinar
TOP	Tonga	Panaga
TRL	Turkey	Turkish Lira (deprecated)
TRY	Turkey	Turkish Lira
TTD	Trinidad and Tobago	Trinidad and Tobago Dollar
TWD	Taiwan	Taiwan Dollar
TZS	Tanzania	Tanzanian Shilling
UAH	Ukraine	Hrynia
UGX	Uganda	Ugandan Shilling
USD	United State of America	US Dollar
UYU	Uruguay	Uruguayan Peso
UZS	Uzbekistan	Som
VEF	Venezuela	Bolivar Fuerte
VND	Vietnam	Vietnamese Dong
VUV	Vanatu	Vatu

Currency Code	Country	Currency
WST	Samoa	Tala
XAF	Central African Republic	Central Africa CFA Franc
XCD		East Caribbean Dollar
XOF		Niger Franc
XPF		New Caledonia Franc
YER	Yemen	Rial
YUM		Yugoslavian Dinar
ZAR	South Africa	South African Rand
ZMW	Zambia	Kwacha
ZWD	Zimbabwe	Zimbabwe Dollar (deprecated)
ZWL	Zimbabwe	Zimbabwe Dollar



When a user's currency preference differs from the currency used in a commerce transaction, it is not possible for the user to reconfigure products or add a catalog item to a commerce document.

Pagination

When 10 or more currencies are being used, fields will be paginated into columns to eliminate horizontal scrolling. Pagination will only occur when five or more currencies are being utilized in AVP Entry Manager, Pricebook Associations, and Customer Specific Pricing (CSP). Five paginated columns will display at a time, each identified by its currency code. Click left/right arrows to load the next/previous set of columns.



The "Currency: Show" drop-down menu allows users to access a currency (or set of currencies) directly without having to navigate through the paginated columns.



In Customer Specific Pricing, any changes that a user makes on a page with pagination will be saved implicitly when switching column views (user must stay within area for the implicit changes to take place).

NOTES



In Pricebook Associations, column headers will repeat at the bottom of the table for reference in long data sets.

Currency menus will display alphabetically by currency code, following the fullaccessuser's base currency.

On the user side: the currency code will be displayed, followed by the display name.

On the fullaccessuser side: the base currency will be highlighted for easy recognition, and precede all other currencies used. When admin hovers over the currency code, a hover tooltip will display the currency name.

Doc Engine includes the ability to format all BigMachines-supported currencies, as long as the symbol is supported by the font.

RELATED TOPICS

SELECT CURRENCY IN TRANSACTION

OVERVIEW

BigMachines allows you to set the transaction currency independent of the logged in user's currency preference. With this feature, the user is able to:

- Create quotes in a currency different from user profile preference.
- Ability to pick session currency different from profile currency at the time of login.
- Change the currency on a transaction.

The Change Currency action is a system-generated main document action, which is automatically generated when a commerce process is created.

Additional Features

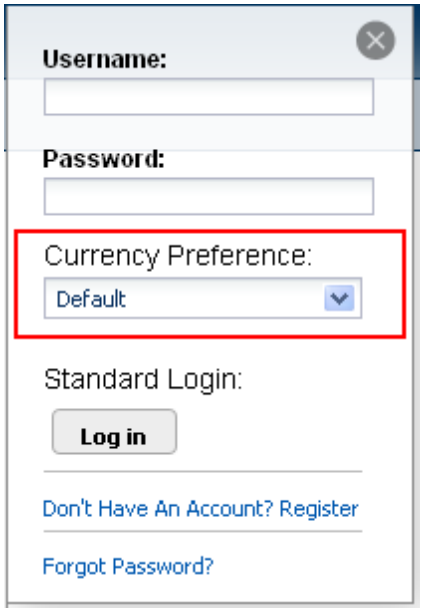
You can set up Integration in a manner that would allow the user currency, transaction currency and price book variable name to be passed in to BigMachines from Salesforce. If the Opportunity currency is different from the user's preferred currency in BigMachines, the commerce document will be created in the currency passed in from Salesforce and not the user's preferred currency. This allows the user to create quotes in a currency different from he/her preferred currency.

Another nice feature provided is that the end user is able to edit all transactions, regardless of their preferred currency. The transaction currency will be applied in configuration if the user chooses to reconfigure or clicks Add from Catalog.

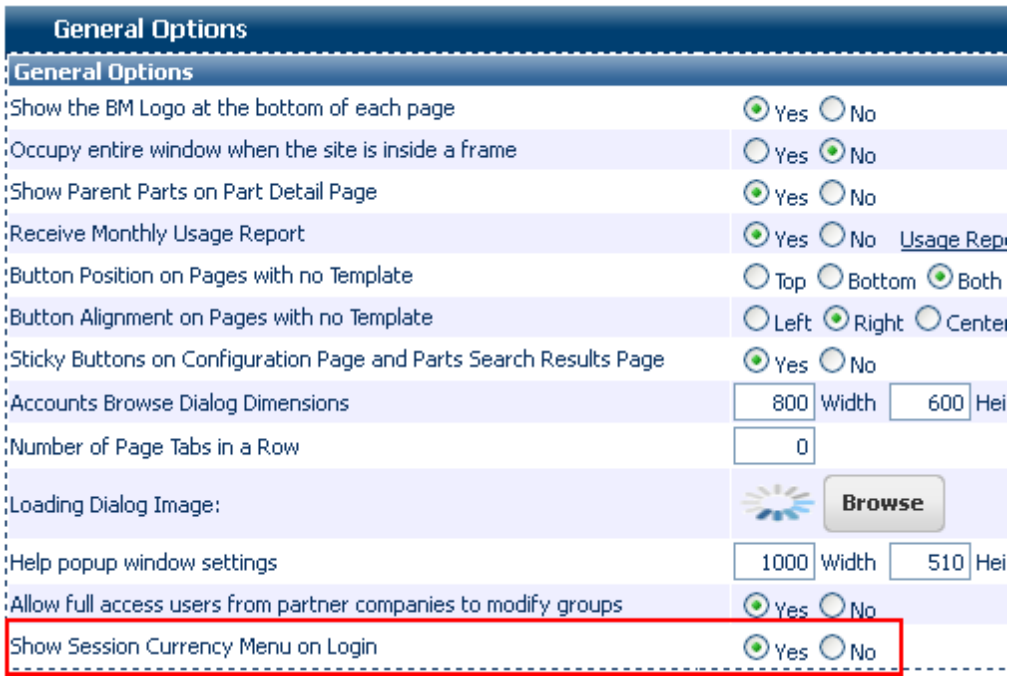
ADMINISTRATION

Choosing Session Currency on Login

For users logging in directly to BigMachines, you can choose to display a session currency drop-down menu on the login page. This menu shows all currencies supported in the application. In this case, the session currency selected by the user on login will be maintained in configuration and commerce. This lets the user to configure and quote in the currency of their choice, regardless of their preferred currency.



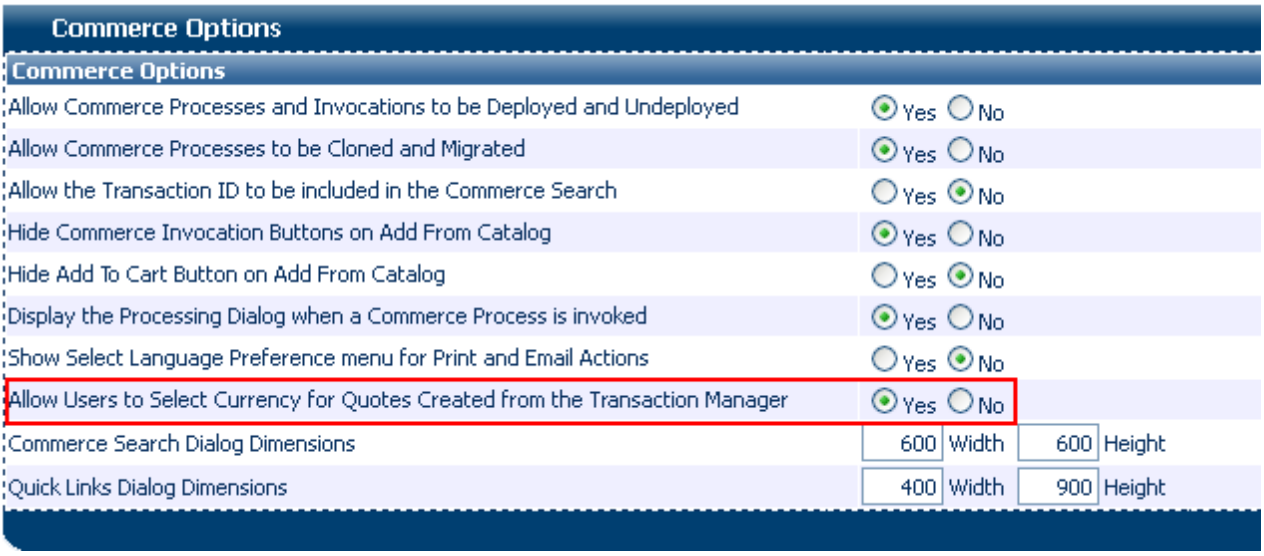
1. Navigate from the Admin Home Page > General > General Site Options.
2. Select **YES** for the option Show Session Currency Menu on Login.



Choosing Transaction Currency on New Transaction Action

If the transaction currency is not passed in through Salesforce, you can choose to display a currency drop-down menu when the end user clicks the **New Transaction** button from the Transaction Manager. The currency picked by the user will be the currency of the transaction.

1. Navigate from the Admin Home Page > Commerce and Documents > Commerce Settings.
2. Select **YES** for the option Allow users to Select Currency for Quotes Created from the Transaction Manager.



Changing Transaction Currency

You can let the end user to change the currency of a transaction that has already been created.

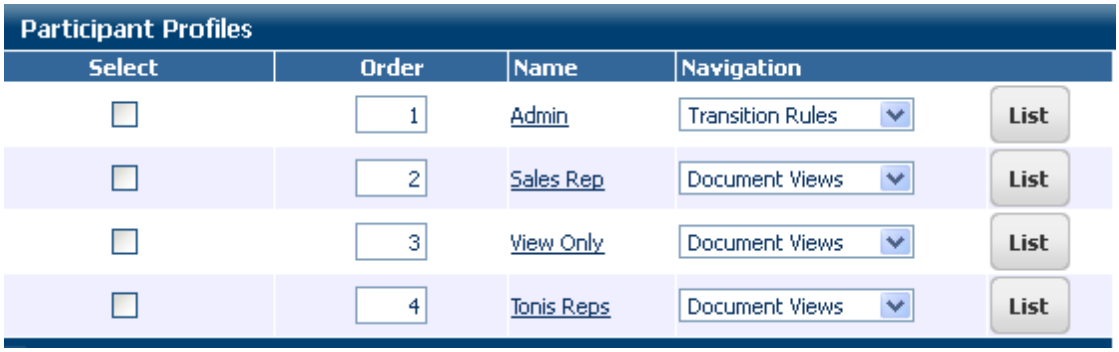
1. Navigate from the Admin Home Page > Commerce and Documents > Process Definition > Select a Commerce Process > Steps.
2. Determine on which Workflow Step(s) the user can make changes to the transaction currency. *In this example, we're using the Pending step.*
3. Select **Participant Profiles** from the drop-down menu and click **List.**



4. Determine which profile(s) should be able to change the transaction currency.

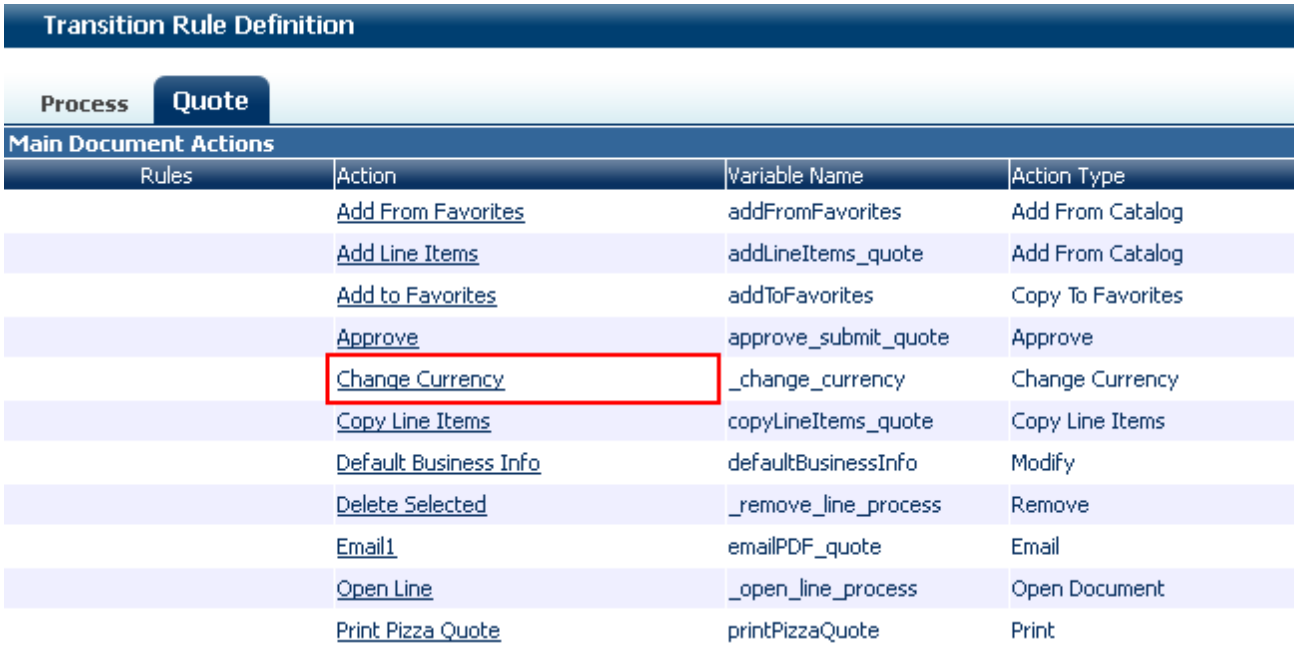
If you'd like to allow more than one profile to change the currency, you'll need to follow the next steps for each profile.

5. Select **Transition Rules** from the drop-down menu and click **List**.



6. Click the **Quote tab** on the Transition Definition Page.

7. Click on the **Change Currency** action.



8. Click **Add** to create a transition rule.

9. Name the rule, provide a description and click **Add**.

10. Click **Save**. The end user will see the following **Change Currency** button and corresponding menu, if there are no line items on the transaction:



NOTES

In commerce, the session currency can be accessed using **_system_user_currency_pref.**

In configuration, the session currency can be accessed using **_BM_USER_CURRENCY.**

Opportunity currency needs to be passed into the parameter **_bm_cm_new_transaction_currency**.

If the change currency action is performed on a document, the exchange rate will be applied to all currency attributes on save. It is recommended that the fullaccessuser specified 'Leave Value Unchanged' for all currency attributes in the Modify tab for the Change Currency action.

BigMachines has removed the limit of concurrent currencies that could be supported per application. Previously limited to 24, user can now enable any and all of the currencies that BigMachines currently supports.

The **_bm_session_currency** is used to set session currency through the punch-in URL.

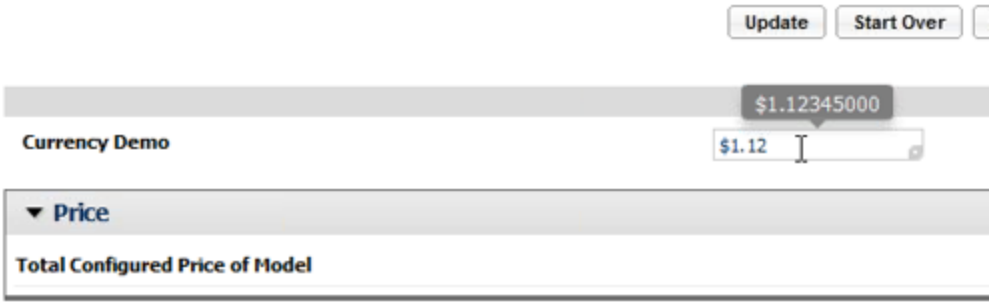
Commerce Integration XSLs will not help in setting the transaction currency equal to the opportunity currency.

RELATED TOPICS

CURRENCY ATTRIBUTE HOVER TOOLTIP

OVERVIEW

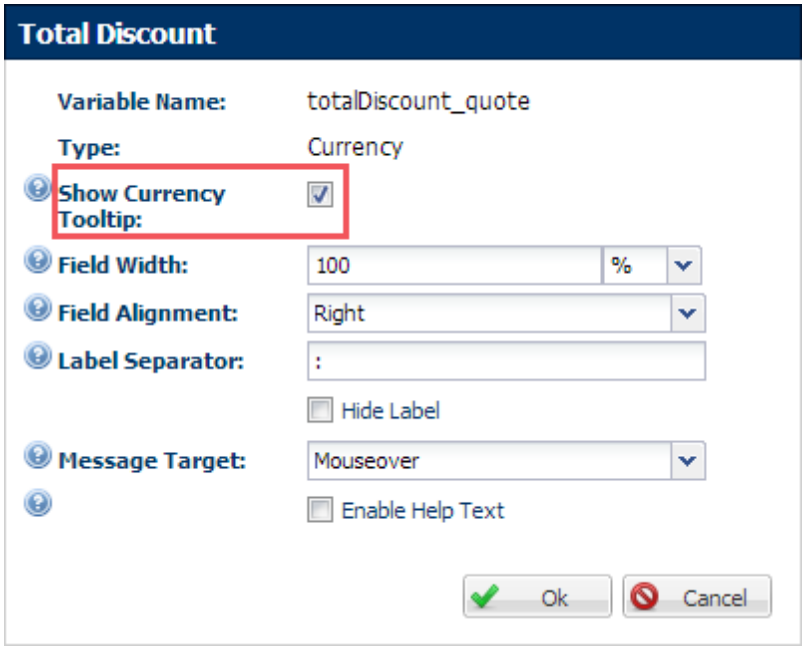
A hover tooltip that will appear if the stored precision value is different than the displayed precision and the numbers are not equivalent. This feature is available only on currency attributes.



ADMINISTRATION

Enable currency attribute hover tool tip in commerce

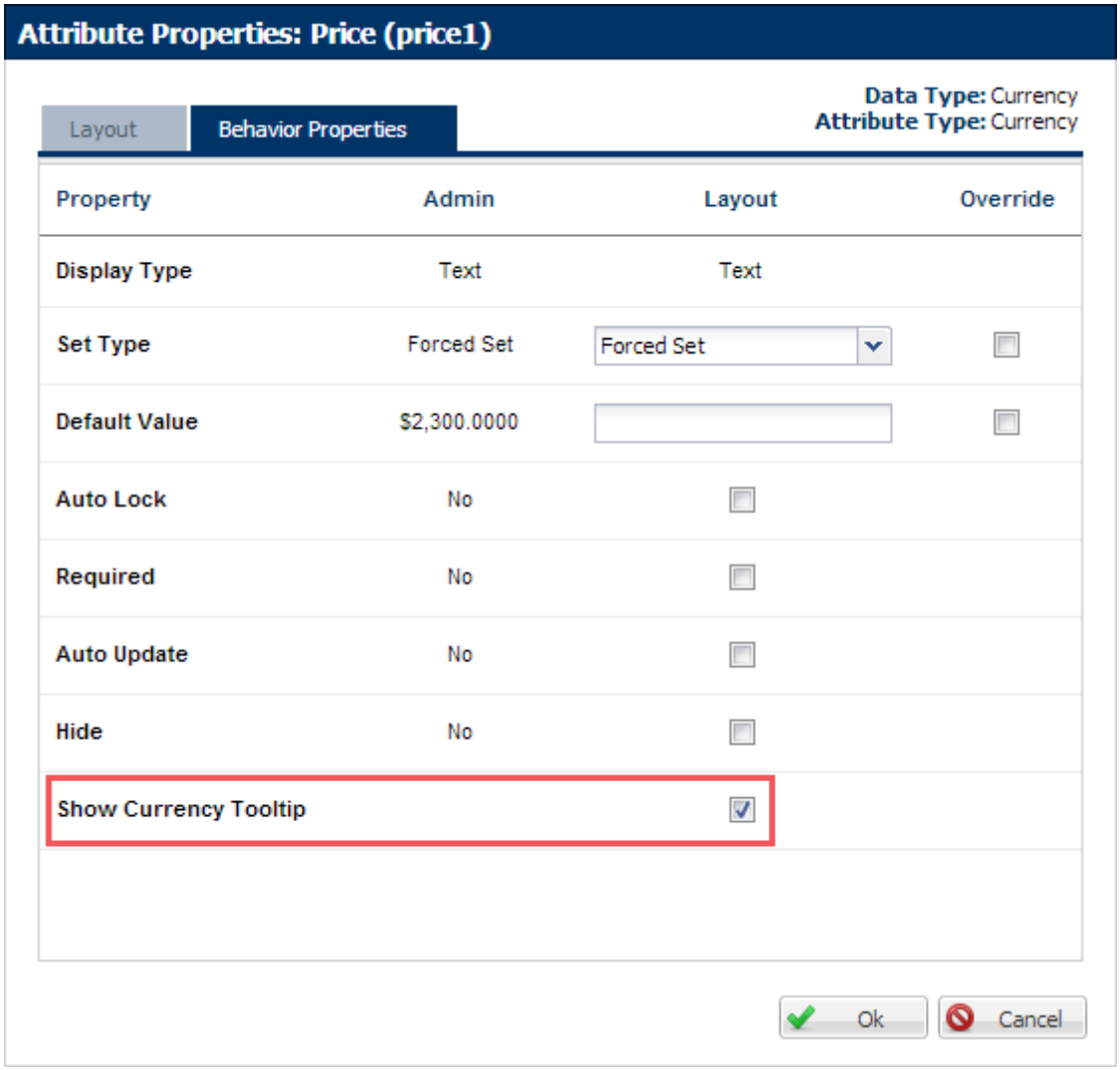
1. Navigate from the Admin Home Page > Process Definition Quick Links > Process > Select a Quote or Line> Select Layout Editor.
2. Click the **edit icon** on a currency attribute.
3. Select the **Show Currency Tooltip** option on the currency attribute editor.



For the Line Item Grid, the condition is for all attributes on the grid.

Enabling currency attribute hover tooltip in configuration

1. Navigate from the Admin Home Page > Catalog Definition Quick Links > Product Family > Find List of Configuration Flow Rules > Select Layout.
2. Click the **edit icon** on a currency attribute.
3. Click the **Behavior Properties** tab.
4. Select the **Show Currency Tooltip** option.



This can also be done on the Behavior Properties tab at the grid level, if there are one or more currency attributes

NOTES



Set default displayed precision, per currency, on the Host Company page.



Decimal precision is available for commerce and/or configuration attributes only, therefore the tooltip is only available on currency attributes.



Displayed precision cannot be greater than what is stored.

RELATED TOPICS

DECIMAL PRECISION

OVERVIEW

Fullaccessusers can set and display higher precision values, up to eight places, for currencies within the application. Many customers have SKU's that are less than a cent; BigMachines allows you to define precision in currency attributes to help maintain smaller price values. You will be able to set decimal place precision for different currencies at the host company level. In addition, individual currency attributes can have their own decimal precision.

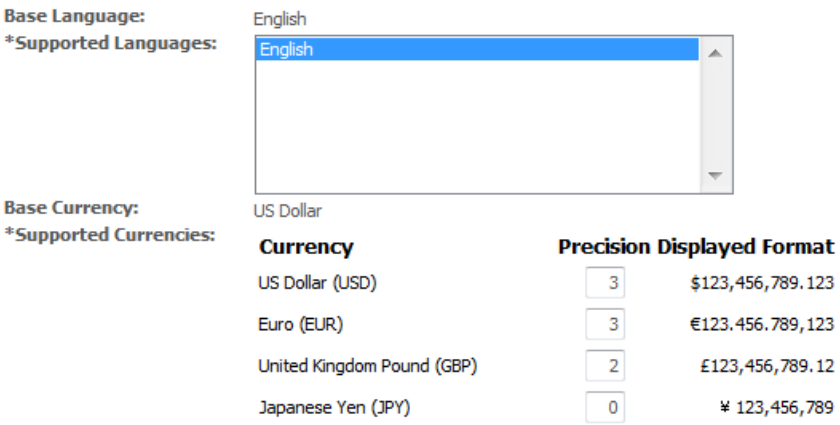
ADMINISTRATION

Setting Decimal Precision Host Company

This section will show you how to set the default display for decimal precision, per currency, at the Host Company Level. **Note:** it is considered best practice that stored and displayed precision be equivalent, unless a discrepancy is expected behavior.

1. Navigate from the Admin Home Page > General > Host Company.
2. Navigate to the **Supported Currencies** section.
3. Set the display precision for each of the supported currencies.

> Displayed precision cannot be greater than what is stored.



4. Click **Apply** or **Update**.

Parts will show the default stored precision of the host company.

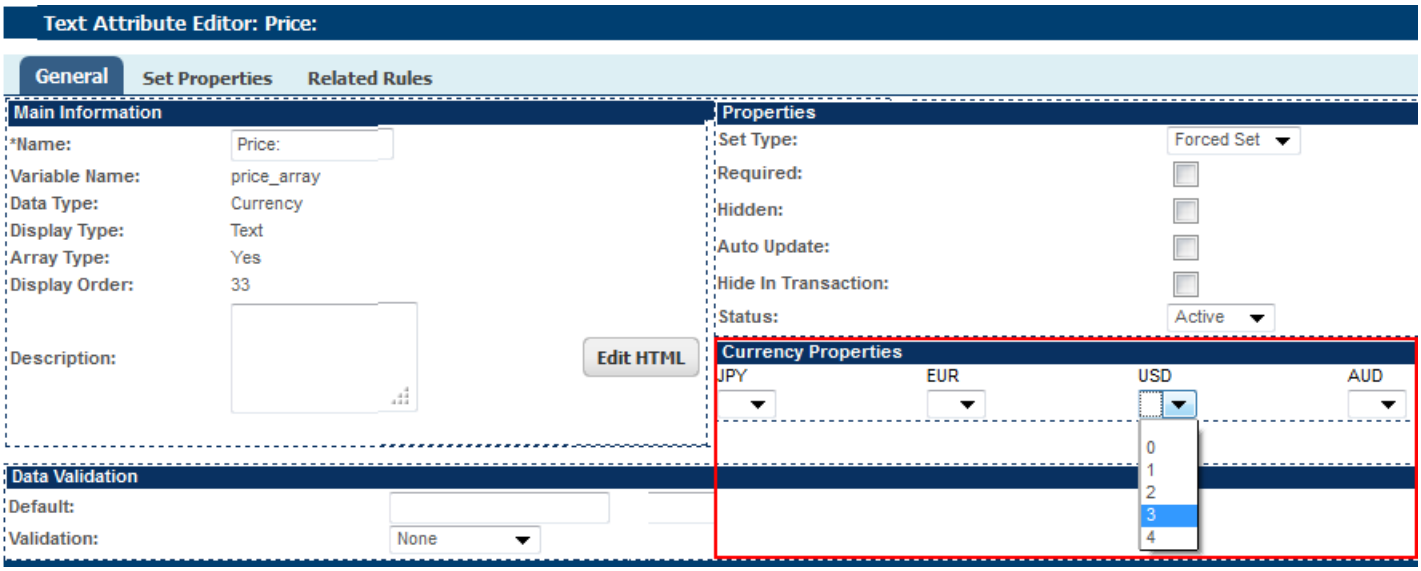
Stored precision values must match in test and production environments in order to migrate anything.

Setting Decimal Precision - Currency Attributes

This section will show you to how to set display for decimal precision, per currency, for commerce and/or configuration currency attributes.

1. Navigate to a commerce or configurable currency attribute.
2. Choose the **decimal precision** for each supported currency under **Currency Properties**.

If this is left blank, display precision will default to the host company settings.



3. Click **Save** or **Update**.



If precision is exceeded upon saving a currency attribute, an error will tell fullaccessusers if precision values need to be changed.

Currency Attribute Hover Tooltip

A hover tooltip will appear if the stored precision is different than the displayed precision and the numbers are not equivalent. This functionality is available only on currency attributes. For more information, see Currency Attribute Hover Tooltip.

USE CASES

Please see the examples below for typical user-side behavior.

Scenario 1: User searches for part "ABC" or configures a model that has a recommended item price.

- The part display page shows the part and price is "1.236"
- The price being carried to commerce (stored value) is "1.2355"
- The Item Price in commerce id "1.24", though the actual value is "1.2355".
- The Final Price is displayed as "1.24"

Scenario 2: User edits a field manually and types in "0.7655" as the value.

Once they click a "Save" action:

- The stored value is "0.7655"
- The Final Price is displayed as "0.77".

Scenario 3: User in Scenario 2 initially tries to enter "0.76555" and click Save, however the system throws an error saying precision is set to 4.

When formatting currency values in the Document Engine, precision will be pulled from the value provided in formatting function.

- If precision value is not mentioned in formatting function, precision of 2 will be displayed.
- The formatting function will not allow the admin to select the display precision on the document.

NOTES



Currency values are saved in stored precision in the line item detail page.

You can view values in the SOAP XML or Printer-Friendly/History XML.

When stored and displayed precision for a part is greater than 2, Salesforce handles a parts export by rounding the values to 2 places.



The system will maintain separate values for stored and displayed precision.

Existing data will not automatically change when precision is increased or decreased.

Rules will fire based on stored precision.

When formatting currency values in the Document Engine, precision will be pulled from the value provided in formatting function.

For parts import and export of precision less than 2, both sides of the integration add trailing zeroes.

On upgrade, the stored precision setting will be set to a value of 2, or the standard for the currency code.



You must contact BigMachines OPS to set stored precision.

Stored precision should not be reduced after it is set. For example, do not reduce precision from 4 decimal places to 2.

Users will not be able to view the stored precision values.

For import, make sure that BigMachines precision is not smaller than Salesforce precision.

RELATED TOPICS

EXCHANGE RATES

OVERVIEW

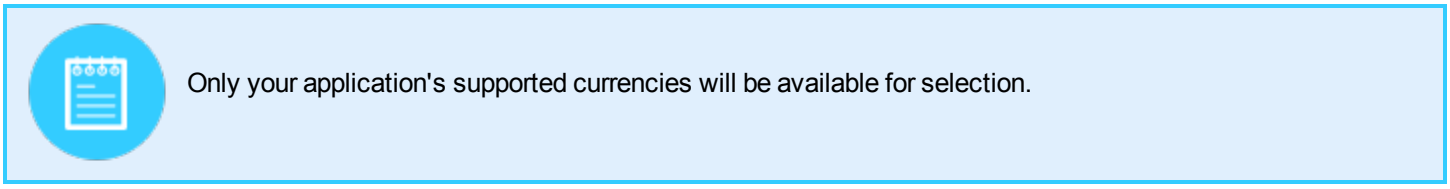
Exchange rates should be set up for all currencies your company supports to ensure that prices are available in the user's currency of choice when the commerce system needs to access them. The commerce system accesses exchange rates when a user's currency preference differs from the currency in which prices are defined.

For example, let's assume that an application supports American Dollars (USD and Euro (EUR). Now, assume that a part A has a price of $500 and that no price is defined in Euro for this part. When a user with EUR as their currency preference tries to buy this part, the system invokes the exchange rate rules to convert the price from USD to EUR. The converted amount is displayed as the price to the buyer. If the part was defined with both a USD and a EUR price, then the system would display the EUR price specified. The exchange rate rules would not be invoked in that case.

ADMINISTRATION

Adding a Currency Conversion (Exchange Rate)

1. Navigate from Admin Home Page > General > General Site Options
2. Click the **Exchange Rates** link.
3. Click **Add**.
4. Choose a **Target Currency** from the drop-down menu.

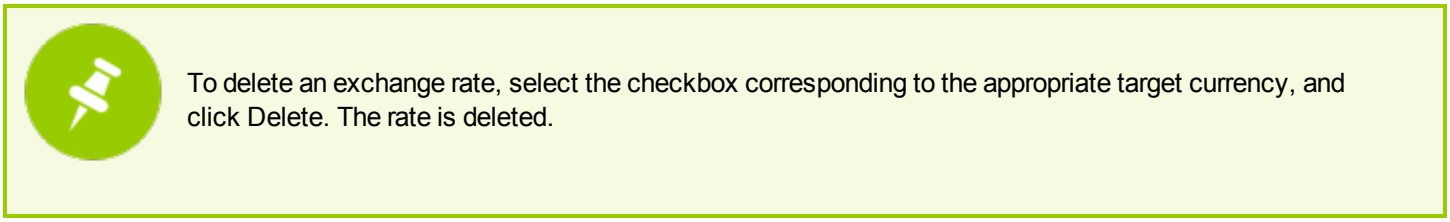


Only your application's supported currencies will be available for selection.

5. Enter a **Conversion Rate**. *This is the existing rate for the currency and can be changed on a daily basis.*
6. Enter an **Uplift Rate**. *This is a multiplier that is used with the conversion rate. It can be used as an administration fee.*



7. Click **Add**.



To delete an exchange rate, select the checkbox corresponding to the appropriate target currency, and click Delete. The rate is deleted.

NOTES



In some cases: it may be appropriate to set this value to 1. Also, note that a value of 0 may break some BML scripts that reference conversion rates in the denominator.



For security purposes, the default conversion rate is set to zero.



If an exchange rate is not available when the system is looking for it, prices are not shown. When an exchange rate is deleted, the system can no longer use it.

RELATED TOPICS

LANGUAGE SUPPORT

OVERVIEW

The system can support multiple languages. Each site must have a base language. Other supported languages can be added to the language set of a site. You can designate any language preference from the available language set. The fullaccessuser with the same language preference as the base language is responsible for providing translations for users that have different language preferences. If translations are not available, then the data is displayed in the base language even for the non-base language user.

Supported Languages for BigMachines 10, 11 and 12+

Language	Language Code
English	en
French	fr
German	de
Spanish	es
Italian	it
Arabic (Saudi Arabia)	ar_SA
Chinese Simplified (China)	zh_CN
Chines Traditional (Taiwan)	zh_TW
Czech	cs_CZ
Danish (Denmark)	da_DK
Dutch (Netherlands)	nl_NL
English (Australia)	en_AU
English (Canada)	en_CA
English (United Kingdom)	en_GB
Finnish	fi_FI
French (Canada)	fr_CA
Greek	el_GR
Hebrew (Israel)	iw_IL
Hindi (India)	hi_IN
Hungarian	hu_HU
Japanese	ja_JP
Korean	ko_KR
Polish	pl_PL
Portuguese (Brazil)	pt_BR
Russian	ru_RU

Language	Language Code
Slovak	sk_SK
Swedish	sv_SE
Thai (Western Digits)	th_TH

Languages Supported with BigMachines 11 and 12+

These languages are in addition to the supported languages above.

Supported Languages: BigMachines 11 & 12+		
ALBANIAN	ENGLISH_NEW_ZEALAND	
ARABIC_ALGERIA	ENGLISH_PHILIPPINES	SERBIAN_SERB_MONT
ARABIC_BAHRAIN	ENGLISH_SINGAPORE	SLOVENIAN
ARABIC_EGYPT	ENGLISH_SOUTH_AFRICA	SPANISH_ARGENTINA
ARABIC_IRAQ	ESTONIAN	SPANISH_BOLIVIA
ARABIC_JORDAN	FRENCH_BELGIUM	SPANISH_CHILE
ARABIC_KUWAIT	FRENCH_LUXEMBOURG	SPANISH_EL_SALVADOR
ARABIC_LEBANON	FRENCH_SWITZERLAND	SPANISH_GUATEMALA
ARABIC_LIBYA	GERMAN_AUSTRIA	SPANISH_HONDURAS
ARABIC_MOROCCO	GERMAN_LUXEMBOURG	SPANISH_MEXICO
ARABIC_OMAN	GERMAN_SWITZERLAND	SPANISH_NICARAGUA
ARABIC_QATAR	GREEK_CYPRUS	SPANISH_PANAMA
ARABIC_SUDAN	ICELANDIC	SPANISH_PARAGUAY
ARABIC_SYRIA	INDONESIAN	SPANISH_PERU
ARABIC_TUNISIA	IRISH	SPANISH_PUERTO_RICO
ARABIC_UAE	ITALIAN_SWITZERLAND	SPANISH_US
ARABIC_YEMEN	JAPANESE_IC	SPANISH_COLOMBIA
BELARUSIAN	LATVIAN	SPANISH_COSTA_RICA
BULGARIAN	LITHUANIAN	SPANISH_DOMINICAN_REPUBLIC
CATALAN	MACEDONIAN	SPANISH_ECUADOR
CHINESE_SIMP_SINGAPORE	MALAY	SPANISH_URUGUAY
CHINESE_TRAD_HK	MALTESE	SPANISH_VENEZUELA
CROATIAN	NORWEGIAN_BOKMAL	THAI_THAIDIGITS
DUTCH_BELGIUM	NORWEGIAN_NYNORSK	TURKISH
ENGLISH_INDIA	PORTUGUESE_PORTUGAL	UKRAINIAN
ENGLISH_IRELAND	ROMANIAN	VIETNAMESE
ENGLISH_MALTA	SERBIAN_BOS_HERZ	

Language Support for Commerce Transactions

The system supports users with a language preference that is different from the language that the quote was created in to view and print translated values for parts description and parts extended description fields. Translations for these fields are available in all languages supported by the application, as long as a host company FullAccess user with the same language preference as the base language of the site has provided these translations.

Once the translations are carried over to the Commerce List, the users can see parts description and extended description fields in the current language (if the translations are present). However, only the user with the same language preference as the one that added the part to the commerce list is able to add the part to the quote.

All the translated values are stored in the Commerce XML file. This XML file is not repopulated with new translation data. The translatable fields in Commerce are:

- Part Description
- Parts Extended Description 1
- Parts Extended Description 2

Sample XML

```
<!--This is the sub document. Assume that the quote was created in the English language and the other languages supported in the site are French and German. Only relevant new tags and there corresponding old tags are mentioned here-->
<document buyer_company_id="175" buyer_user_id="82" data_type="2" document_id="586" document_number="2" supplier_company_id="8175">

<!--normal part desc tag. Assuming quote to be in English Language-->
<_part_desc>englishDescription</_part_desc>

<!--new part description itn tag. Entry for the transaction language is not added in the new tag as its already available in the main tag-->
<_part_desc_itn>
<fr>frenchDescription</fr>
<de>germanDescription</de>
</_part_desc_itn>
<_part_extended_desc_1>engExtendedDescription1</_part_extended_desc_1>

<!-- new itn tag for extended desc -->
<_part_extended_desc_1_itn>
<fr>FrenchExtendedDescription1</fr>
<de>germanExtendedDescription1</de>
</_part_extended_desc_1_itn>
<_part_extended_desc_2> englishExtendedDescription2</_part_extended_desc_2>

<!-- new itn tag for extended desc -->
<_part_extended_desc_2_itn>
```

```
<fr>frenchExtendedDescription2</fr>
<de>germanExtendedDescription2</de>
</_part_extended_desc_2_itn>
</document>
```

ADMINISTRATION

Translations Bundle

Working with a professional translation company, BigMachines has translated site content into the following languages:

- French
- German
- Spanish

All product templates and pre-existing BigMachines content users receive at implementation is available in translation.



The addition of the translations bundle does not affect current customer functionality.

User is not required to make any selections, once site language preference established, customer site will appear with translations in place.

Other Translations

BigMachines is currently fully available in English, French, German, Russian, Chinese and Japanese. This includes all labels and admin pages working in those languages. To enable further on-screen labels to be translated, ensure that the Text Administration template for that language has been implemented. Contact your Project Manager if this is not the case.

- **Attributes:** Attribute label translations are available under the Translation button on the attribute's admin page. Here you can enter the text for all of the languages supported on that site.



To translate an attribute, you must first be managing BigMachines as an English language user.

- **Menu values:** In the case of Menu attributes, the text for the UI for the menu options, can be translated with the Entry Manager button. The menu items must be created first.
- **Attribute Help page translations:** The text that appears in a new window when a user presses (?) next to a configuration attribute, can be adapted to the user's language by translating the description text (and Long Description when available).

- **Home Page options:** The configuration model names can be adapted to the supported languages, via the Home Page Admin site. Expand the Product Family, Product Line and enter the Model punch-in. In that admin sub-screen, you will find a Translations button. DO not forget to re-deploy the home page to reflect changes.
    - The Product Line description text that appears on the home page can be translated by navigating to the Product Line admin screen (Product Line Administration List, click on the name itself). The product line name itself is not translatable, but the label can appear differently on the Home Page.
    - The Product Family Name is changed under the Configuration Product Definition admin screen. Click on the Family name, and within the next screen a Translations button is available. This latter option will require the Product Family to be re-deployed.

Reconfigure in different language

Users can reconfigure a model in a language other than the quote language, which is determined by the quote creator's language preferences. Configuration Rules will utilize the language of the user performing the Reconfigure; Configuration Flow Rules will utilize the original specified language. Rules will only behave differently if the Configuration Rule inputs depend on the system language Attribute.

Export Attachment action language popup

Export Attachment-type actions are located the "Show Select Language Preference menu for Print and Email actions" Commerce setting. Once an attachment is exported in Commerce, a popup dialog will appear allowing the user to choose a system-selected language to export to the CRM. In order to take advantage of this feature, the document must use the system-selected language attribute; the selected language defaults to the user-preferred language.



Like other actions, the document will not be automatically translated.

NOTES



Administrators can retrieve a specific translation of a Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. *If a language value is set, that value will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.*

RELATED TOPICS

LOCALIZATION

OVERVIEW

BigMachines helps you prepare your application for an international roll-out by allowing you to manage all of your translations through Bulk Upload and Download. From here, you can download all commerce and configuration translations in CSV format, have someone translate it, and finally re-upload it to BigMachines. You will be able to filter by language and BigMachines module (currently Commerce or Configuration).

ADMINISTRATION

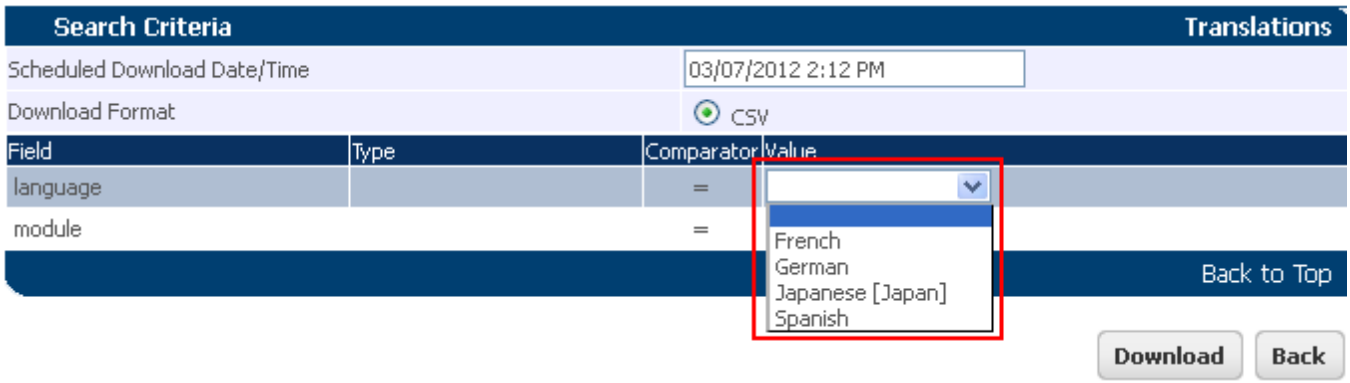
The typical use case will be to download the translation file(s) first.

Downloading Translations

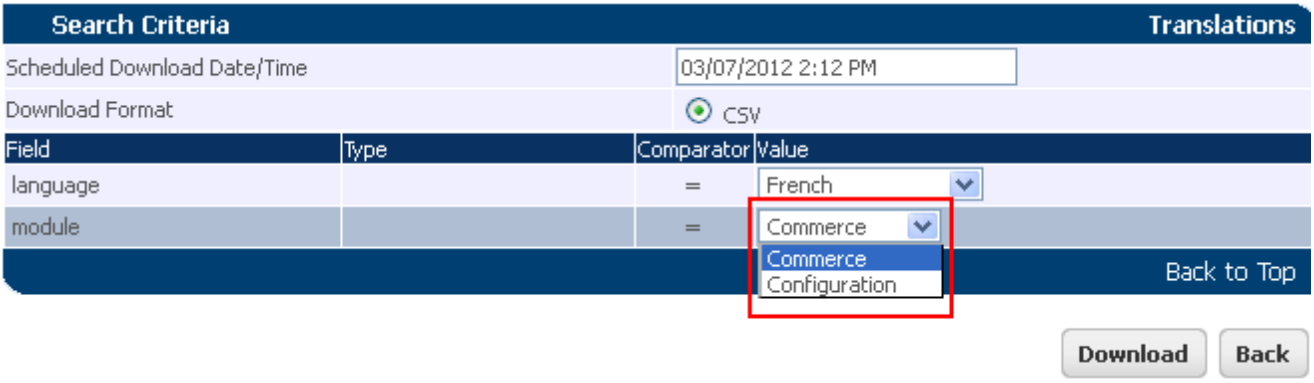
1. Navigate from Admin Homepage > Utilities > Bulk Download.
2. Select the **Translations radio button** then click **Next**.

> Move on to Step 5 if you do NOT need to filter your results based on language or BigMachines module.

3. Choose the **Language** you would like to download from the language drop-down menu.

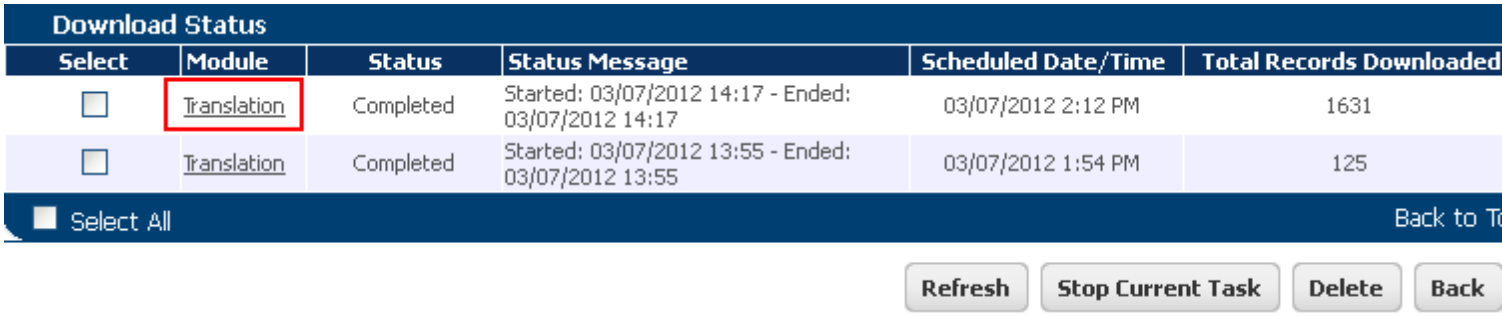


4. Choose a BigMachines **module(s)** from the drop-down menu: Commerce, Configuration, Part and Part Extended [Descriptions], Part Associations, Part Custom Fields, Serial Number, Serial Number Custom Fields and Price Books.



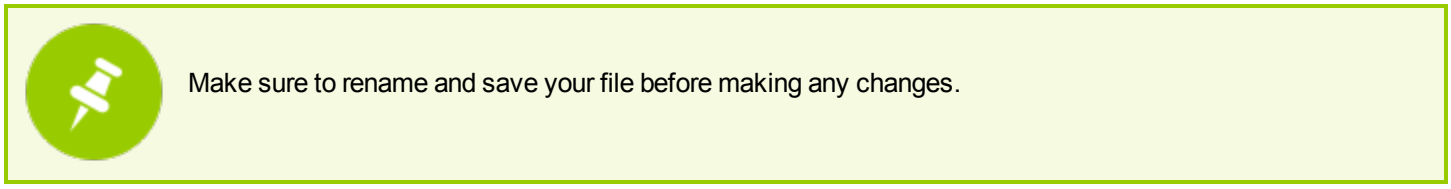
5. Click **Download**.

6. Click the **Translation** file from the Download Status window.



7. **Unzip** (or Save and then Unzip) the file.

8. Double-click on the **CSV** file to open.



9. Translate your content. You will notice that there will be a column for the module, a contextual aid to assist the translator (so they know how it should be translated), the element being translated (is it a label, description, etc.), as well as what needs to be translated.

   Example of Commerce:

	D	E	F	G	H	
1						
2						
3	module	context	reftype	type	en	fr
4	Commerce	SalesEngine C	CmProcess	Process Nam	SalesEngine Commerce Process	
5	Commerce	SalesEngine C	CmProcess	Description	- Feature includes:	1) prix par volume 2) groupes de tra
6	Commerce	SalesEngine C	CmProcessColu	Label	Quote Number	
7	Commerce	SalesEngine C	CmProcessColu	Label	Account Name	
8	Commerce	SalesEngine C	CmProcessColu	Label	Revision	
9	Commerce	SalesEngine C	CmProcessColu	Label	Description	
10	Commerce	SalesEngine C	CmProcessColu	Label	Status	
11	Commerce	SalesEngine C	CmProcessColu	Label	Prepared By	
12	Commerce	SalesEngine C	CmAction	Label	Forward	
13	Commerce	SalesEngine C	CmAction	Description	Used to forwards documents to other users.	
14	Commerce	SalesEngine C	CmAction	Label	New Transaction	
15	Commerce	SalesEngine C	CmAction	Description	Used to create a new transaction with no line items.	
16	Commerce	SalesEngine C	CmAction	Label	Print	
17	Commerce	SalesEngine C	CmAction	Description	Generates documents in printer friendly format	
18	Commerce	SalesEngine C	CmAction	Label	Open Quote	

Example of Configuration:

	A	B	C	D	E	F	G	H	I
1	Translations								
2									
3	update	refid	fieldno	module	context	reftype	type	en	fr
4	add	5106911	0	Configuration	All Product Family > Ar	PrdFamily	Label	Amo la Pizza di Chicago	
5	add	5106963	0	Configuration	All Product Family > Ar	ConfigAttr	Name	Name	
6	add	5106963	2	Configuration	All Product Family > Ar	ConfigAttr	Description	Registered catalog item.	
20	add	5107016	0	Configuration	All Product Family > Ar	ConfigAttr	Name	Size	taille
21	add	5107016	2	Configuration	All Product Family > Ar	ConfigAttr	Description		
22	add	5107018	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	Small	petit
23	add	5107019	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	Medium	moyen
24	add	5107020	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	Large	grand
25	add	5107021	0	Configuration	All Product Family > Ar	ConfigAttr	Name	Crust Type	Type de la croûte
26	add	5107021	2	Configuration	All Product Family > Ar	ConfigAttr	Description		
27	add	5107023	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	Thin Crust	croûte fine
28	add	5107024	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	Deep Dish	croûte profonde
29	add	5107025	0	Configuration	All Product Family > Ar	ConfigAttr	Name	Specialty	
30	add	5107025	2	Configuration	All Product Family > Ar	ConfigAttr	Description		
31	add	5107027	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	Meat Lovers	
32	add	5107028	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	The Works	
33	add	5107703	0	Configuration	All Product Family > Ar	MenuItem	Menu Item Text	Create Your Own	

Column Header	What Does It Mean?	What Does It Tell You?
update	Update	This is the action that will be performed when you bulk upload the translated file.**This should not be modified.**
refid	Reference ID	This is the unique ID that BigMachines generates for each item it stores. This could be a rule, attribute, commerce process,

Column Header	What Does It Mean?	What Does It Tell You?
		etc. **This should not be modified.**
fieldno	Field Number	This is another ID that BigMachines generates for each item it stores. **This should not be modified.**
module	BigMachines Module	This tells you which module of the application the terms apply to. For example: commerce, configuration, parts, etc**. This should not be modified.**
context	Context	This provides a navigation path to let you know where the term is being used. For example: *All Product Family > Amo la Pizza di Chicago > Size > Small*.**This should not be modified.**
reftype	Reference Type	This tells you what is being referenced within BigMachines. For example, process columns, commerce actions, commerce attributes, product family, configurable attribute, etc. **This should not be modified.**
type	Type	This tells you what you the type of content are translating. It could be a label, description, default data, menu item text, etc. **This should not be modified.**
en	English Terms	This provides you with the English terms that need to be translated. **This should not be modified.**
fr, esp, etc.	Translated Terms	This will show you any terms that have already been translated in the selected language. This is where you'll add or update your translations.

Example: fr

update	refid	fieldno	module	context	reftype	type	en	fr
add	5106911	0	Configuration	All Product Family > Amo la Pizza di Chicago	PrdFamily	Label	Amo la Pizza di Chicago	Amo la Pizza di Chicago
add	5106963	0	Configuration	All Product Family > Amo la Pizza di Chicago > Size	ConfigAttr	Name	Size	
add	5106963	2	Configuration	All Product Family > Amo la Pizza di Chicago > Small	MenuItem	Menu Item Text	Small	Petit

---

Using Japanese & Chinese Characters

Users should have the East Asian Languages Fonts installed on their laptops or computers to see a predictable behavior with these steps or when working with characters of these languages in Excel, notepad or even browser.



Even if it shows up on the browser does not mean that the fonts are installed.

Verifying if fonts are installed (for Windows XP)

1. Go to Control Panel
2. Open Regional and Language Options
3. Go to the Languages Tab

4. Verify that Installed Files for East Asian Languages is checked. If not, please talk to IT OPS to get it installed before proceeding.

Opening an existing CSV file containing CJK (Chinese, Japanese, Korean) characters (Excel 2007)

1. Open Excel (not the .CSV file)
2. Go to the Data Tab
3. In the Get External Data section in the toolbar, choose from text and select the translation file.
4. On the first wizard screen, select the File Origin menu value to be UTF-8.
5. After going through the Wizard file, the data will show correctly.

Selecting UTF-8 encoding for new files

1. Go to File > Excel Options
2. Select Advanced on the left-side panel.:
3. Click on the Web Options button.
4. Go to the encoding tab.
5. Select 'Save this document as "UTF-8"'.

NOTES



Translations will not affect table data. This functionality allows you to upload translations for all the data that has been customized in the application (attributes, attribute values, rules, etc.).

Standard upload actions not available: add, modify, delete, and delete all.



Please keep in mind that the translations available for download are only content that has been created by a fullaccessuser. This includes the product hierarchy, configurable attributes, commerce processes, commerce actions, etc.

Configuration and/or Commerce must be deployed to see changes.

RELATED TOPICS

COUNTRY & STATE DETAILS

OVERVIEW

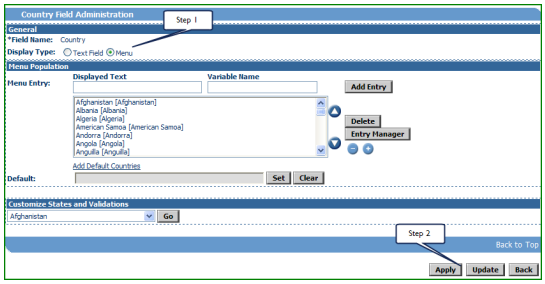
Here you can manage country and state attributes throughout the site. The countries added here will be displayed on the User, Company and Account administration pages. They will also get deployed with the commerce processes and will be displayed on country type fields on the user side transactions.

Navigation: Admin Home Page > Global System Settings and Utilities > Countries and States

ADMINISTRATION

Setting the Display Type

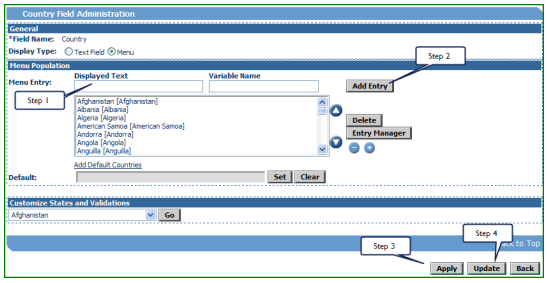
1. Click **Text Field** or **Menu** to determine the **Display Type**.
    - **Text Field:** displays a message saying the change will delete all the existing Menu entries on Save, and that by selecting the option, you may also introduce some legacy data issues on the user-side.
    - **Menu:** displays the options as a drop-down list menu.



2. Click **Apply** to save your changes and remain on the current page.

Adding a Country Name

1. Enter a name in the **Displayed Text** field in Menu Entry.
2. Click **Add**. *The new name is displayed at the bottom of the list of the displayed countries.*



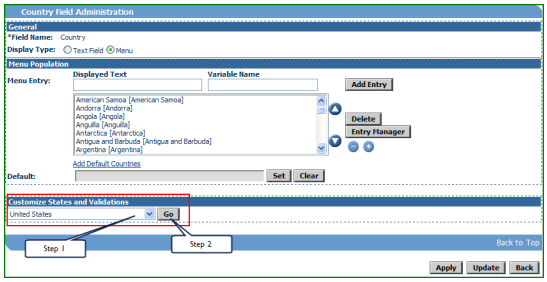
3. Use the Up/Down buttons to shift the country name in the list.

4. Click **Apply** to save your changes and remain on the current page.
5. Click **Update** to save your changes and return to the administration home page.

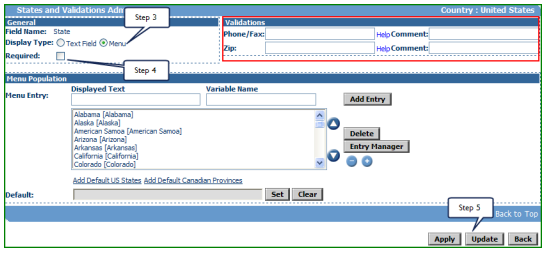
Customizing States & Validations

This section is used to specify validations for the phone, fax and zip information for a selected country. The validation is provided as a Java Regular Expression.

1. Choose a name from the drop-down menu under the **Customize States and Validations** section.
2. Click **Go** to launch the State and Validations Administration page.



3. Click **Text Field** or **Menu** to determine the Display Type.
4. Select the checkbox if the Validation should be **Required**.
    - Provide comments and validations for the phone/fax and zip fields. *Validations should be in the form of a Java Regular Expression.*
    - Click the Help link next to the fields for more information on Java Regular Expressions.
5. Click **Update** to save your changes.

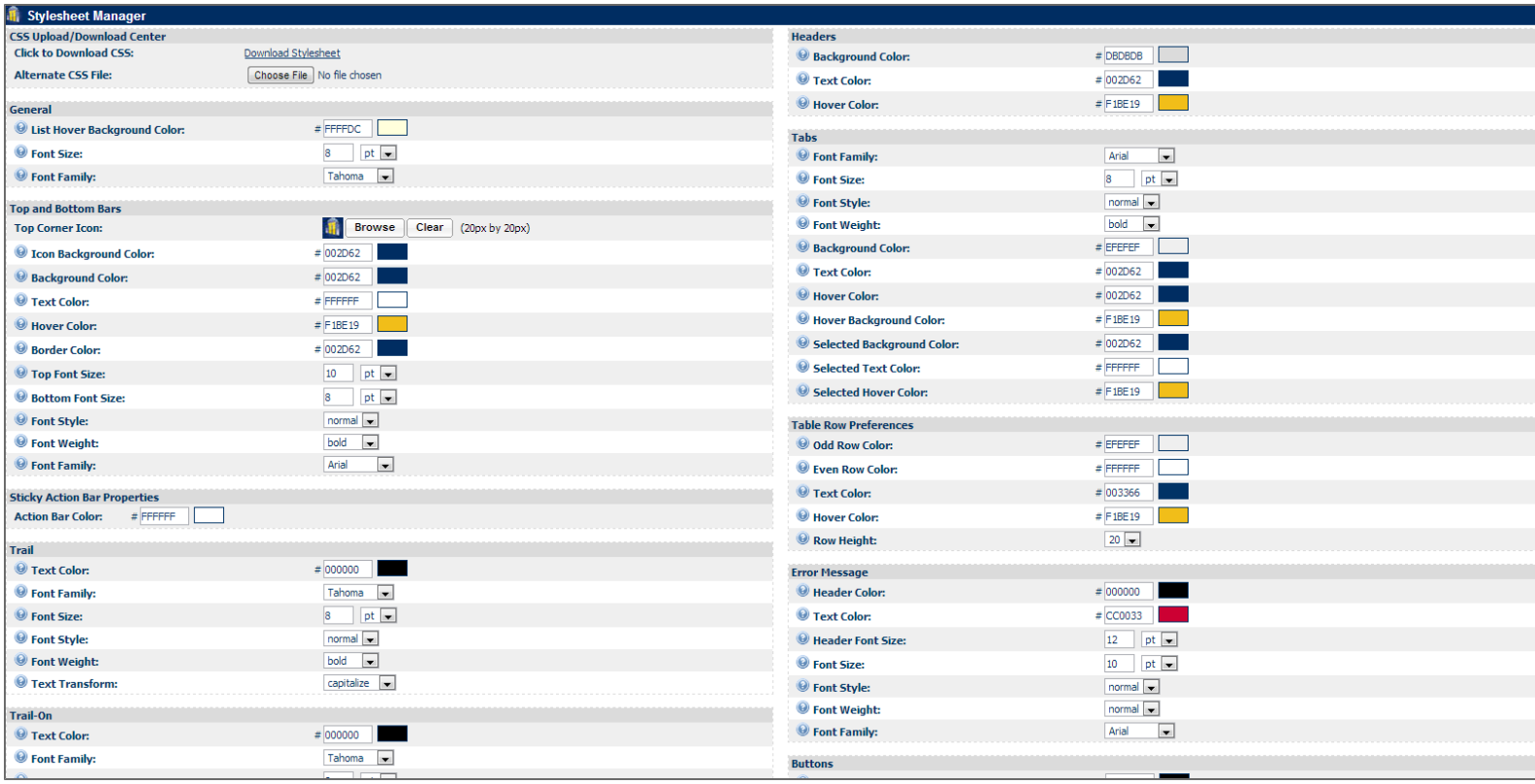


RELATED TOPICS

STYLESHEET MANAGER

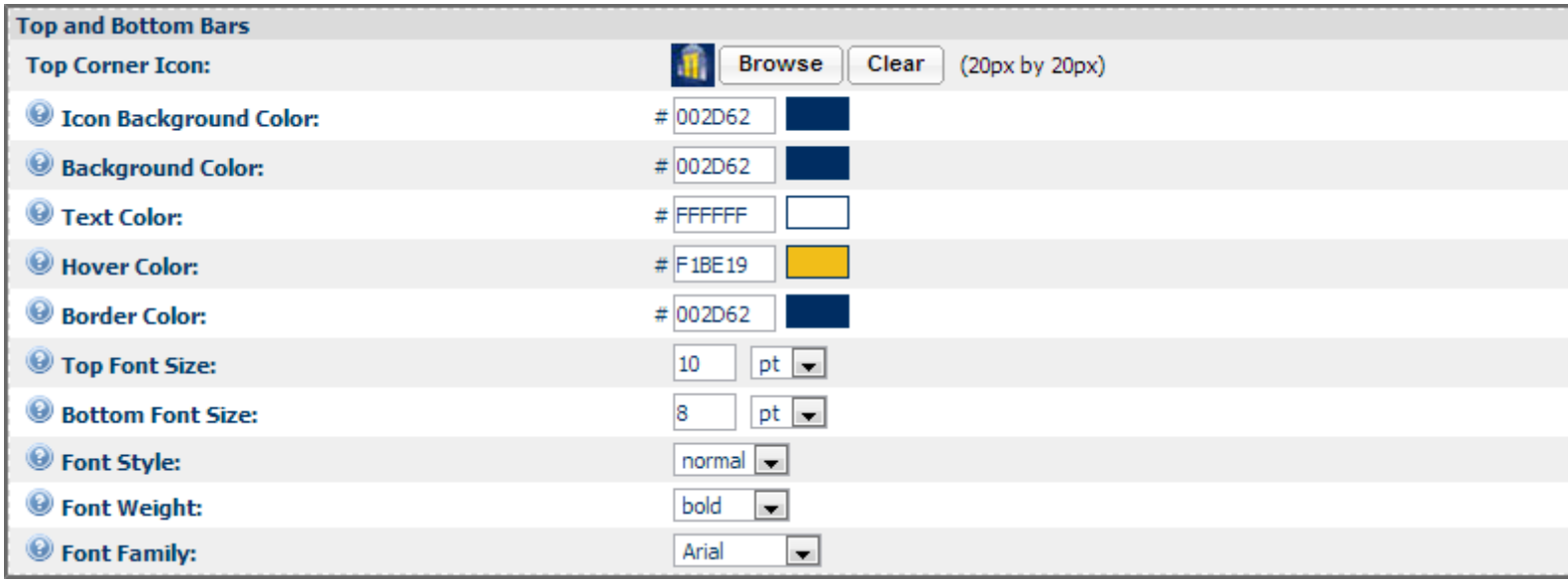
OVERVIEW

The Stylesheet Manager enables you to customize and modify the BigMachines system's font, colors, and images so that they closely match your company's colors and style. A CSS file is used to generate the uniform look and feel of the web site. A default CSS file is provided with the system. The UI provides an easy way to change the colors and fonts used. For advanced customization, you can download the default CSS file provided, make changes and then upload it as an alternate CSS file.



Top & Bottom Bars

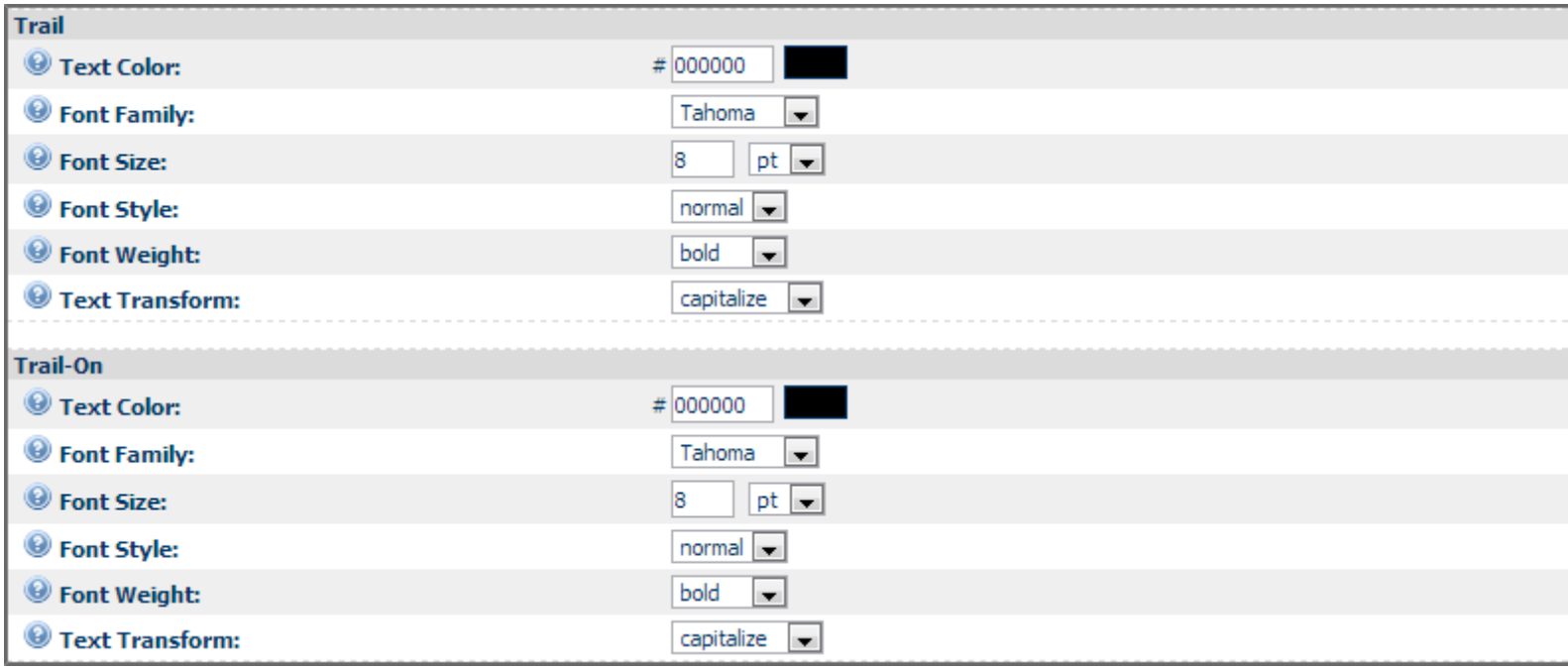
Top and bottom bars frame page content. Typically, each page has a single top bar and a single bottom bar. On the **Branding** page, **Stylesheet Manager** displays in the top bar and **Back to Top** displays in the bottom bar.



Setting	Description
Icon Background Color	Specify here the background color of the icon. You can see a preview of the color for the value entered.
Background Color	Specify here the background color of the bars. You can see a preview of the color for the value entered.
Text Color	Specify here the color of the text that will be displayed on the bars. You can see a preview of the color for the value entered.
Hover Color	Specify here the hover color. This is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.
Border Color	Specify here the border color of the bars. You can see a preview of the color for the value entered.
Top Font Size	Using the drop-down list provided, select the font size of the text that will be displayed on the top bar.
Bottom Font Size	Using the drop-down list provided, select the font size of the text that will be displayed on the bottom bar.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed on the bars.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed on the bars.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed on the bars. The default font is Tahoma.

Trail & Trail-On

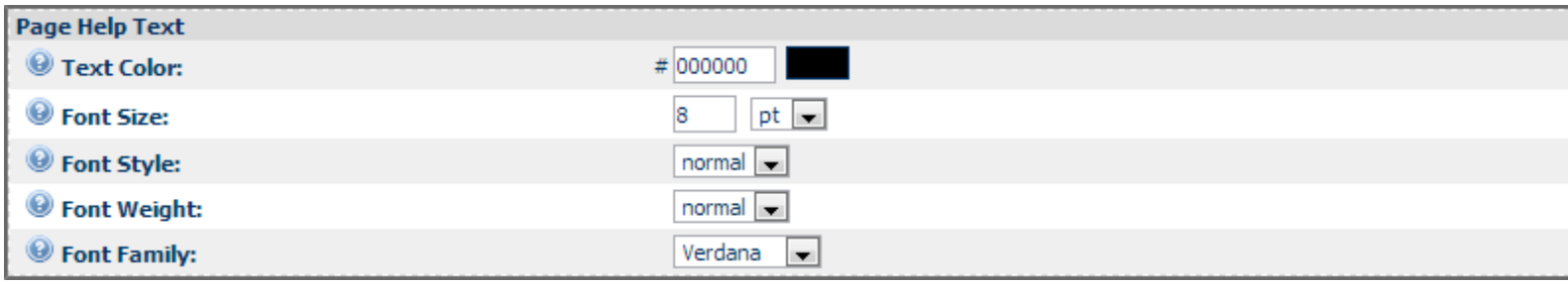
Trail provides a path at the top of the page to indicate the current position of the user in the site structure. **Trail-On** is the portion of the trail where the user is currently at.



Setting	Description
Text Color	Specify here the color of the text that will be displayed. You can see a preview of the color for the value entered.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed. The default font is Tahoma.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed.
Text Transform	Using the drop-down list provided, select whether the text would be capitalized, upper case, lower case, or none.

Page Help Text

Using this section you can customize the look and feel of the help text.

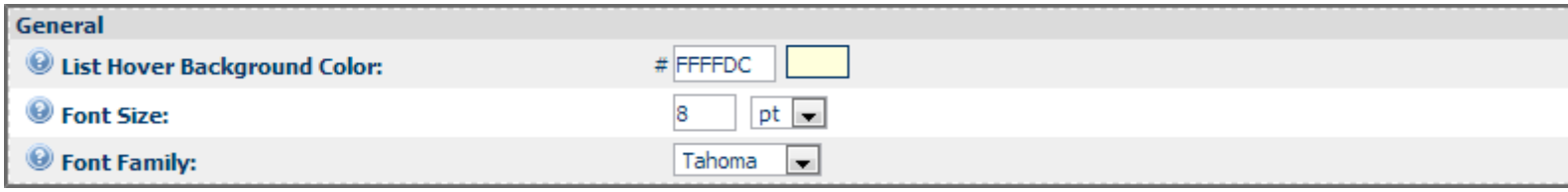


Setting	Description
Text Color	Specify here the color of the text that will be displayed. You can see a preview of the color for the value entered.

Setting	Description
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed. The default font is Tahoma.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed.

General

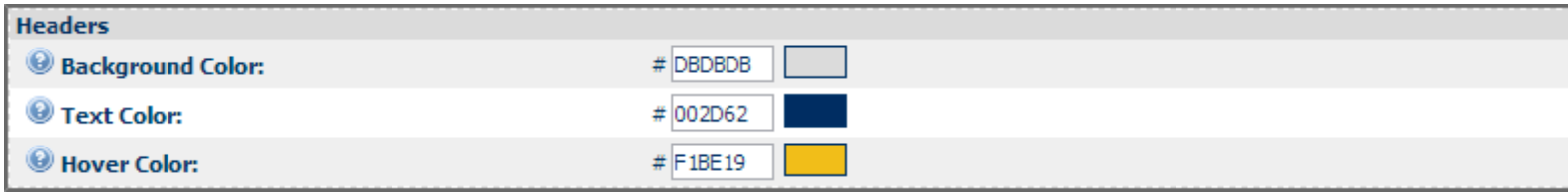
Using this section, you can customize the general look and feel of the website.



Setting	Description
List Hover Background Color	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed. The default font is Tahoma.

Headers

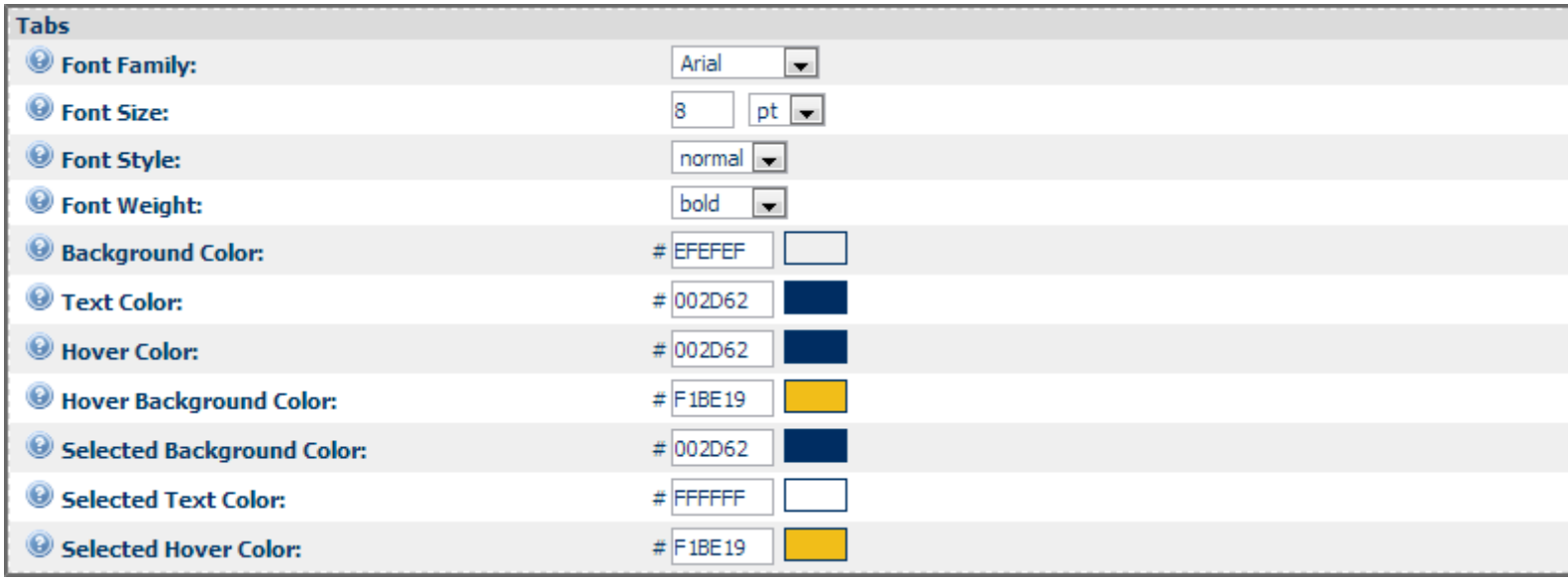
Commerce pages typically have multiple sections. Headers mark the start of a section. **Example:** on the Branding page, a few headers are: Top and Bottom Bars, Headers, Table Row Preferences, and Buttons.



Setting	Description
Background Color	Specify here the background color of the headers that mark the start of a section. You can see a preview of the color for the value entered.
Text Color	Specify here the color of the text that will be displayed on the headers that mark the start of a section. You can see a preview of the color for the value entered.
Hover Color	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.

Tabs

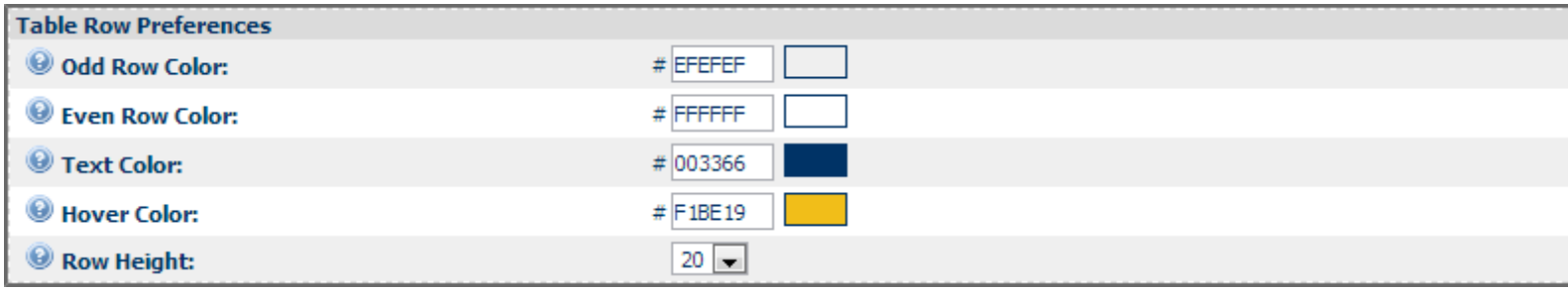
Tabs are used to provide structure to a web page by logically dividing the material in a page.



Setting	Description
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed.

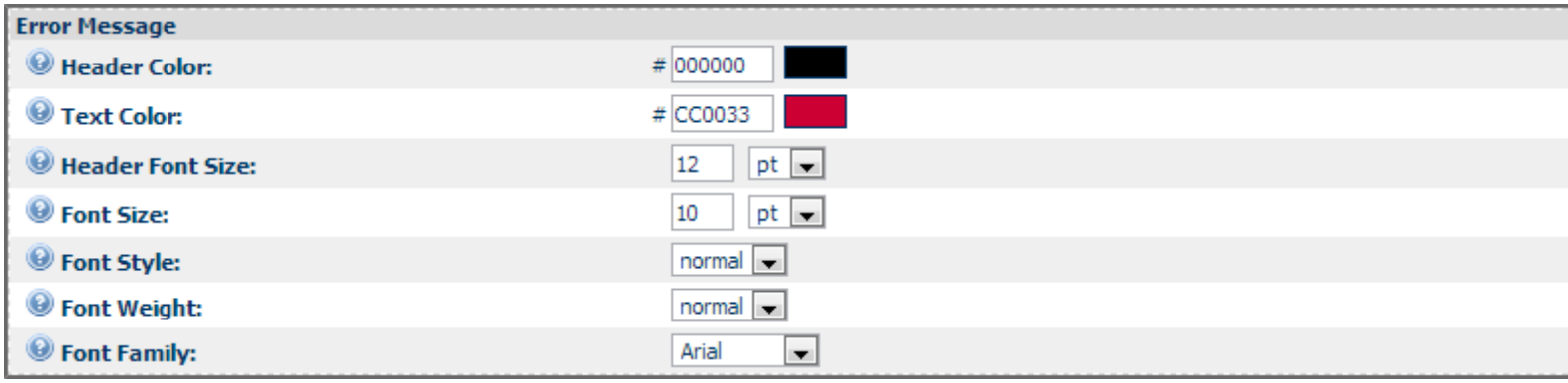
Table Row Preferences

Much of the data that displays on commerce pages is displayed in tables. You can alternate row colors to make tables more reader-friendly. You can also distinguish table text with a custom color.



Setting	Description
Odd Row Color	Specify here the background color of the odd-numbered rows of the table. You can see a preview of the color for the value entered.
Even Row Color	Specify here the background color of the even-numbered rows of the table. You can see a preview of the color for the value entered.
Text Color	Specify here the color of the text that will be displayed on the bars. You can see a preview of the color for the value entered.
Hover Color	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.
Row Height	Using the drop-down list provided, select the height of the rows of the table.

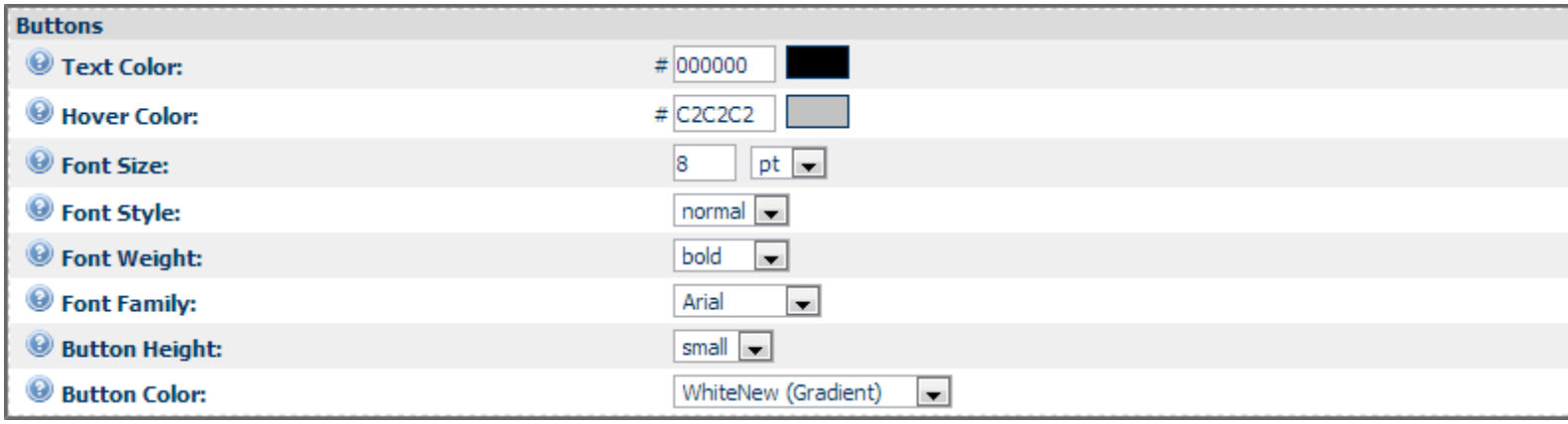
Error Message



Setting	Description
Header Color	Specify here the color of the error message header.
Text Color	Specify here the color of the error message text that will be displayed. You can see a preview of the color for the value entered.
Header Font Size	Using the drop-down list provided, select the font size of the error message header.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed in the error message.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed in the error message.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed in the error message.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed in the error message. The default font is Tahoma.

Buttons

Buttons enable users to perform operations. You can customize the background, text color, and sizes of all the buttons on your site.



Setting	Description
Text Color	Specify here the color of the text that will be displayed on the buttons. You can see a preview of the color for the value entered.
Hover	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the

Setting	Description
Color	color for the value entered.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed on the buttons.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed on the buttons.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed on the buttons.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed on the buttons. The default font is Tahoma.
Button Height	Using the drop-down list provided, select the height of the button.
Button Color	Using the drop-down list provided, select the color of the buttons.

ADMINISTRATION

Customizing your Stylesheet

1. Specify the fonts and color palette of your choice.
2. Do one of the following:
   - **To preview settings:** Click Preview. The refreshed view of the Site Branding page displays the new settings. To adjust the settings, make changes and then click Preview. When the page appears the way you want it to look, click Accept.
   - **To accept settings:** Click Accept.



Branding changes take effect immediately after clicking Accept. If the changes do not appear on a page, it is likely the browser is displaying a cached copy of the page. When this happens, click the browser's Refresh button to update the page.

When using top navigation for your site, the sidebar background color should be set to white (FFFFFF) and the text should be a dark, high-contrast color.

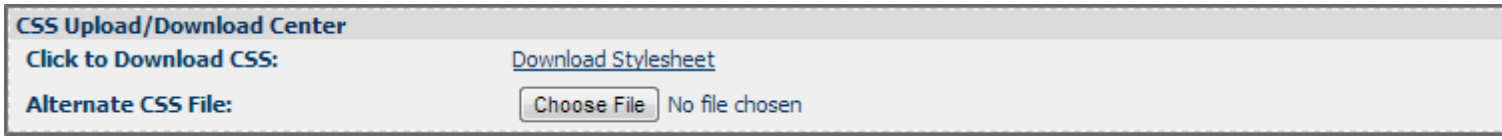


Only websafe colors can be selected from the color palette popup. RGB color codes display in the text-box next to the color preview box. *For non-websafe colors, you can enter RGB color codes and the corresponding color will be shown in the color preview box.*

Downloading a CSS File

This file is useful for developing XSL template files as it enables you to view the classes you can use in XSL.

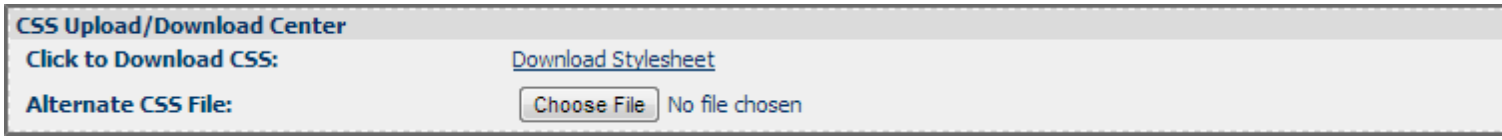
1. Click the **Download Stylesheet** link.



2. Save the downloaded CSS file, modify it, and then update it as an alternative CSS file.

Uploading a CSS File

1. Click **Browse** in the **Alternative CSS Information**. *This displays the File Upload dialog box.*

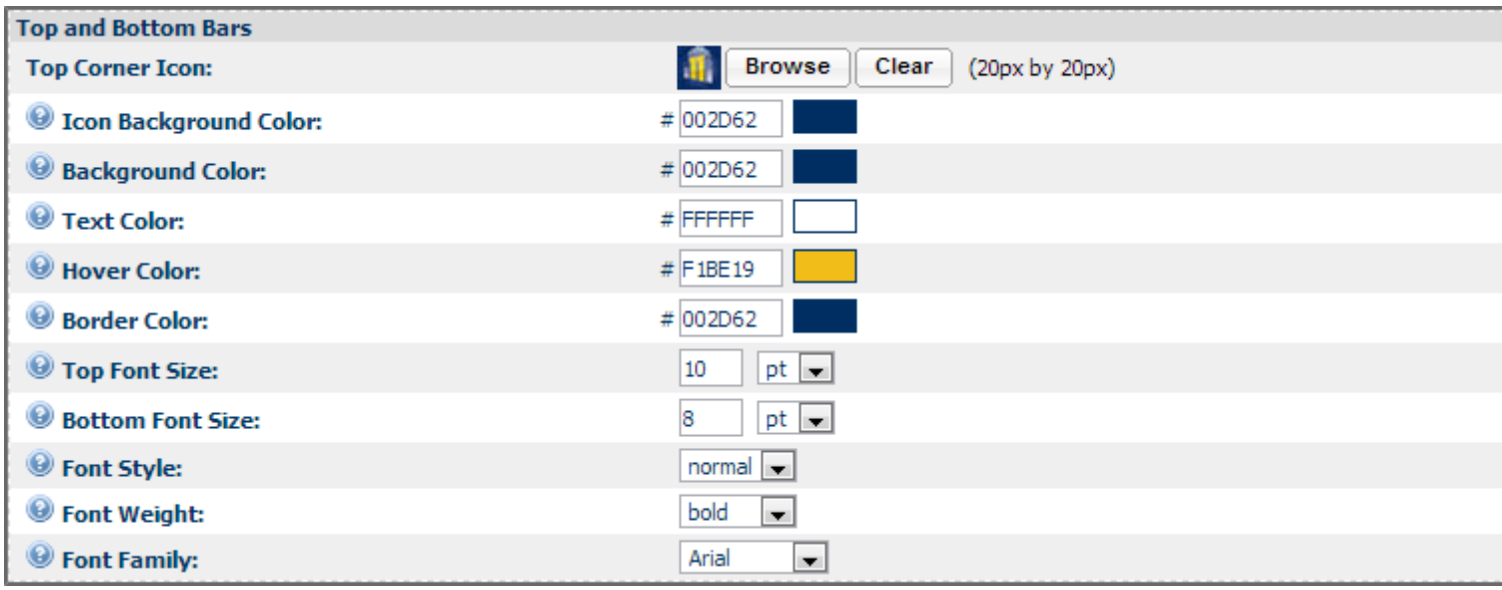


2. Browse for, select the required file, and click **Open**. This displays the selected file name for the **Alternative CSS Information** field.

3. Click **Accept** on the **Stylesheet Manager** page.

Specifying Icons for Top & Bottom Bars

1. Click **Browse** within the **Top Corner Icon** bar to search for and select a new logo. *This will open a new window, File Manager-Browse.*



2. Select the folder from which you desire to import the new logo, using the **Folder** drop-down list. *Selecting a folder name displays the logo images present in the folder, along with a Preview, Description, and the date and time the logo was imported to the folder.*

3. Click Select the entry for the desired logo. *This closes the **File Manager - Browse** window, and displays the name of the selected image in the **Home Page Set Up**.*

4. Click **Clear** to remove the selected icon. *The maximum size allowed for the icon is 20px by 20px.*

NOTES



By default, the sidebar contains the site's navigation links. Alternatively, navigation links can also be displayed across the top of the page. If you use sidebar navigation, then you can customize the sidebar's background color, text color and background image. *The width of a sidebar image should be a maximum of 150 pixels.*

It is recommended that you select one font to use across your entire site; it lends consistency to your site. *The default font is Tahoma.*

The drop-down list displays websafe colors.



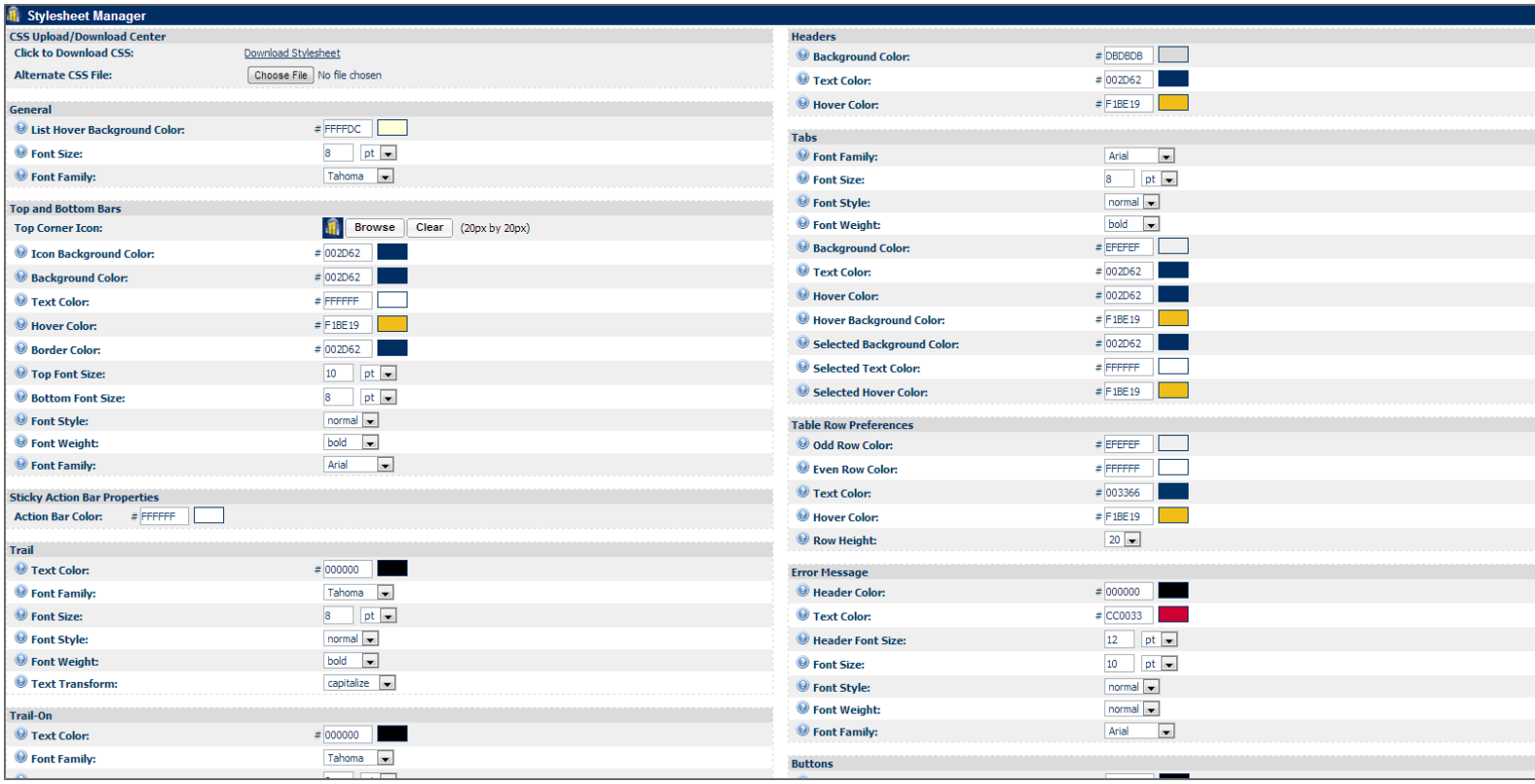
Information regarding design principles:
-Smashing Magazine
-UX Design
-Yahoo UX
-Lynda.com
-Eric A. Meyer Blog
-A List Apart
-W3C Markup Validation Service
-QuirksMode.org

RELATED TOPICS

STYLESHEET MANAGER

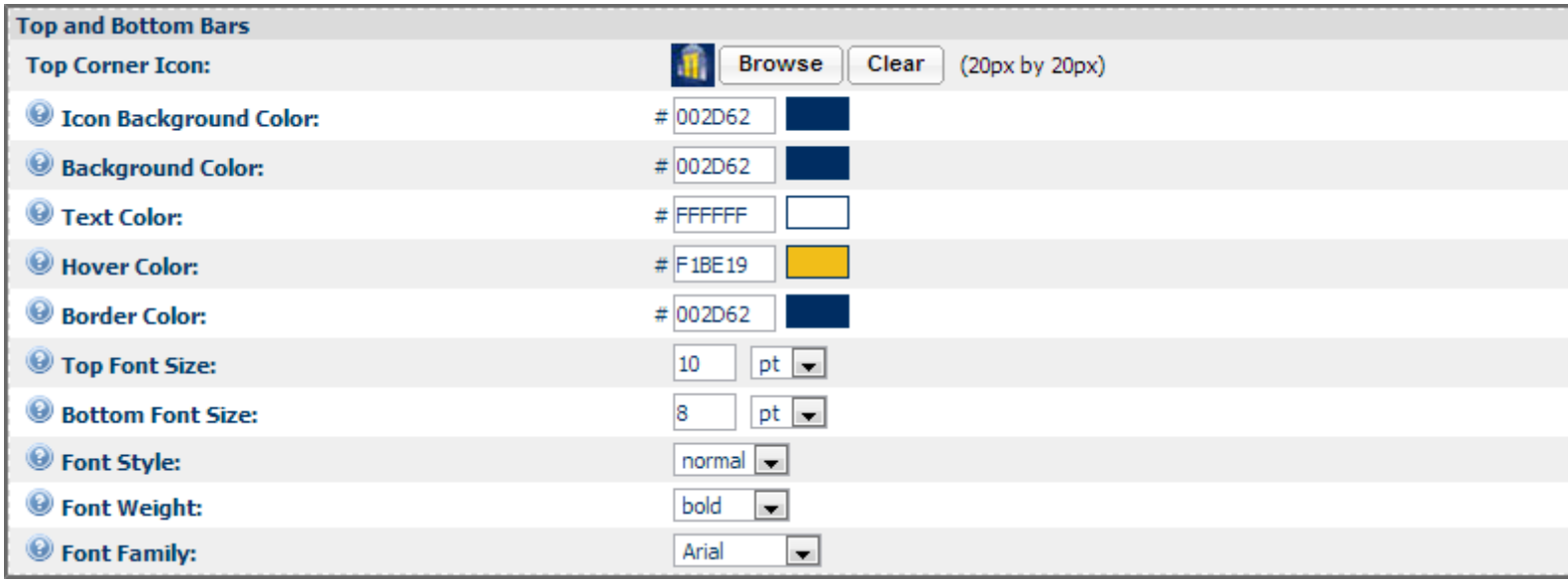
OVERVIEW

The Stylesheet Manager enables you to customize and modify the BigMachines system's font, colors, and images so that they closely match your company's colors and style. A CSS file is used to generate the uniform look and feel of the web site. A default CSS file is provided with the system. The UI provides an easy way to change the colors and fonts used. For advanced customization, you can download the default CSS file provided, make changes and then upload it as an alternate CSS file.



Top & Bottom Bars

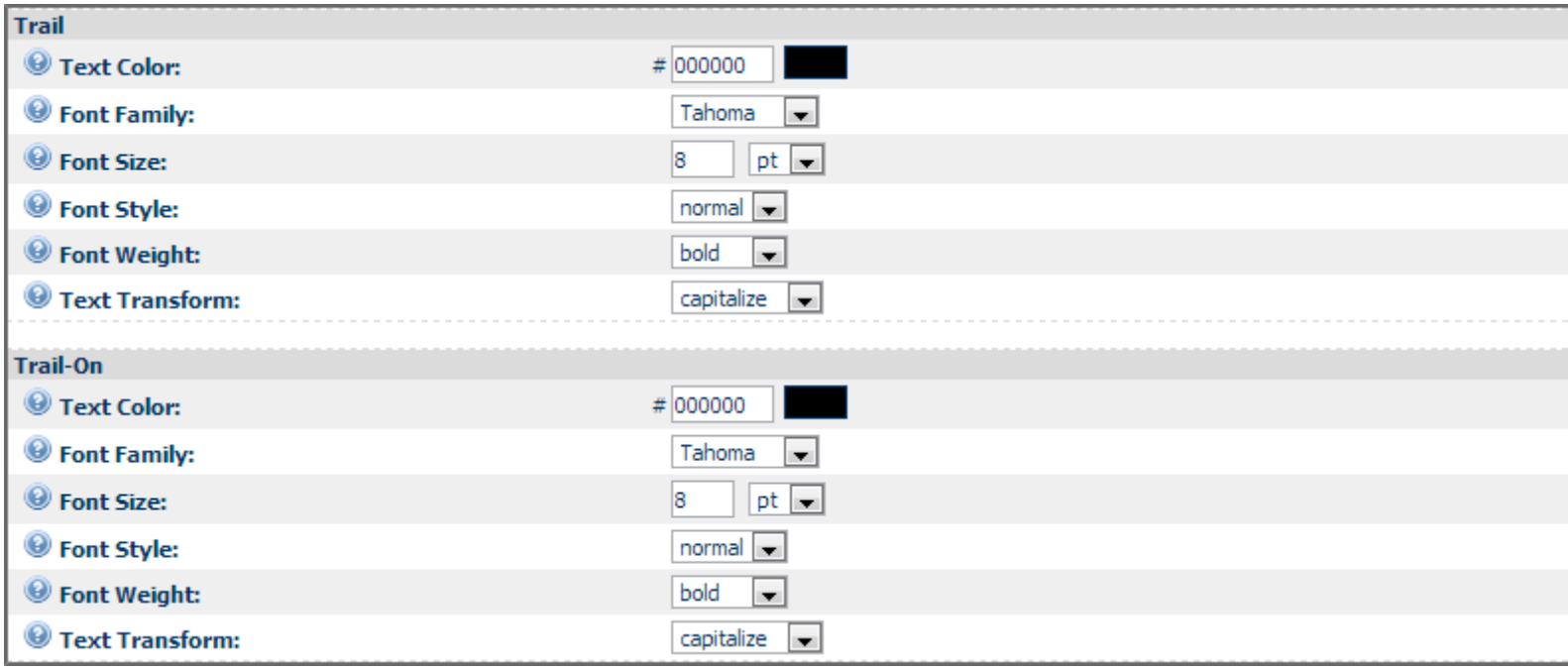
Top and bottom bars frame page content. Typically, each page has a single top bar and a single bottom bar. On the **Branding** page, **Stylesheet Manager** displays in the top bar and **Back to Top** displays in the bottom bar.



Setting	Description
Icon Background Color	Specify here the background color of the icon. You can see a preview of the color for the value entered.
Background Color	Specify here the background color of the bars. You can see a preview of the color for the value entered.
Text Color	Specify here the color of the text that will be displayed on the bars. You can see a preview of the color for the value entered.
Hover Color	Specify here the hover color. This is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.
Border Color	Specify here the border color of the bars. You can see a preview of the color for the value entered.
Top Font Size	Using the drop-down list provided, select the font size of the text that will be displayed on the top bar.
Bottom Font Size	Using the drop-down list provided, select the font size of the text that will be displayed on the bottom bar.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed on the bars.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed on the bars.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed on the bars. The default font is Tahoma.

Trail & Trail-On

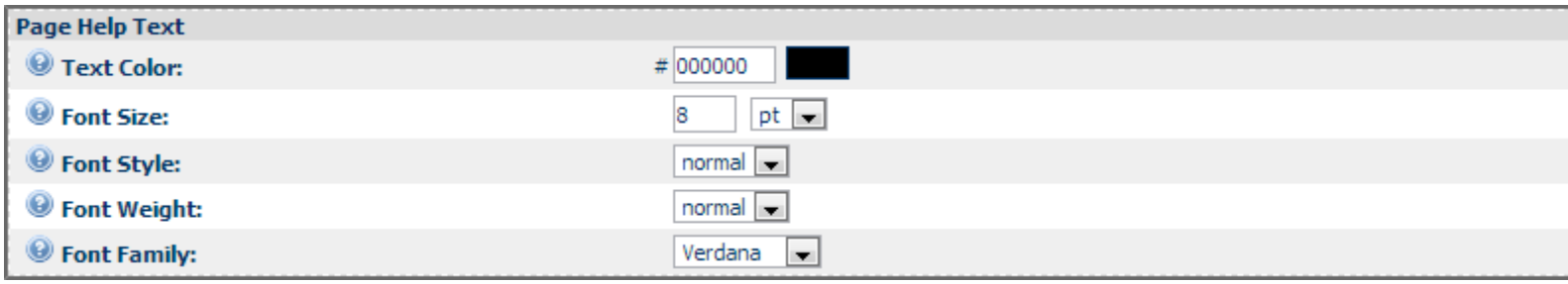
Trail provides a path at the top of the page to indicate the current position of the user in the site structure. **Trail-On** is the portion of the trail where the user is currently at.



Setting	Description
Text Color	Specify here the color of the text that will be displayed. You can see a preview of the color for the value entered.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed. The default font is Tahoma.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed.
Text Transform	Using the drop-down list provided, select whether the text would be capitalized, upper case, lower case, or none.

Page Help Text

Using this section you can customize the look and feel of the help text.

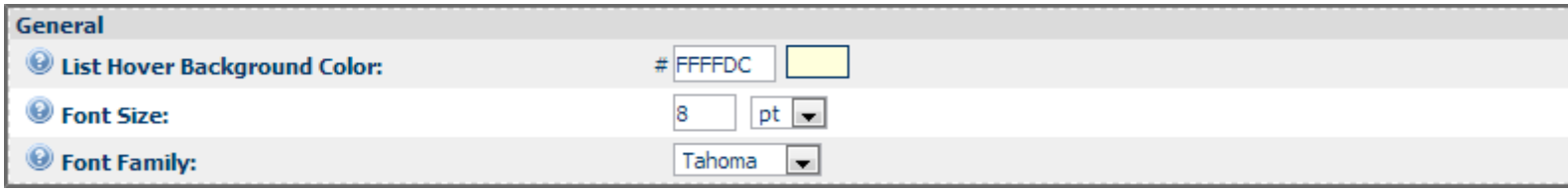


Setting	Description
Text Color	Specify here the color of the text that will be displayed. You can see a preview of the color for the value entered.

Setting	Description
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed. The default font is Tahoma.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed.

General

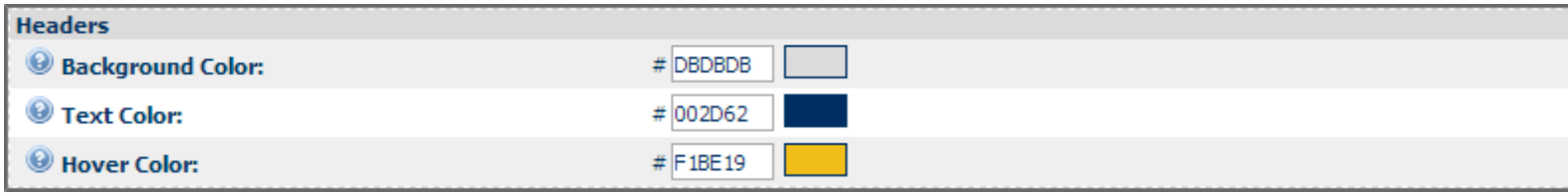
Using this section, you can customize the general look and feel of the website.



Setting	Description
List Hover Background Color	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed. The default font is Tahoma.

Headers

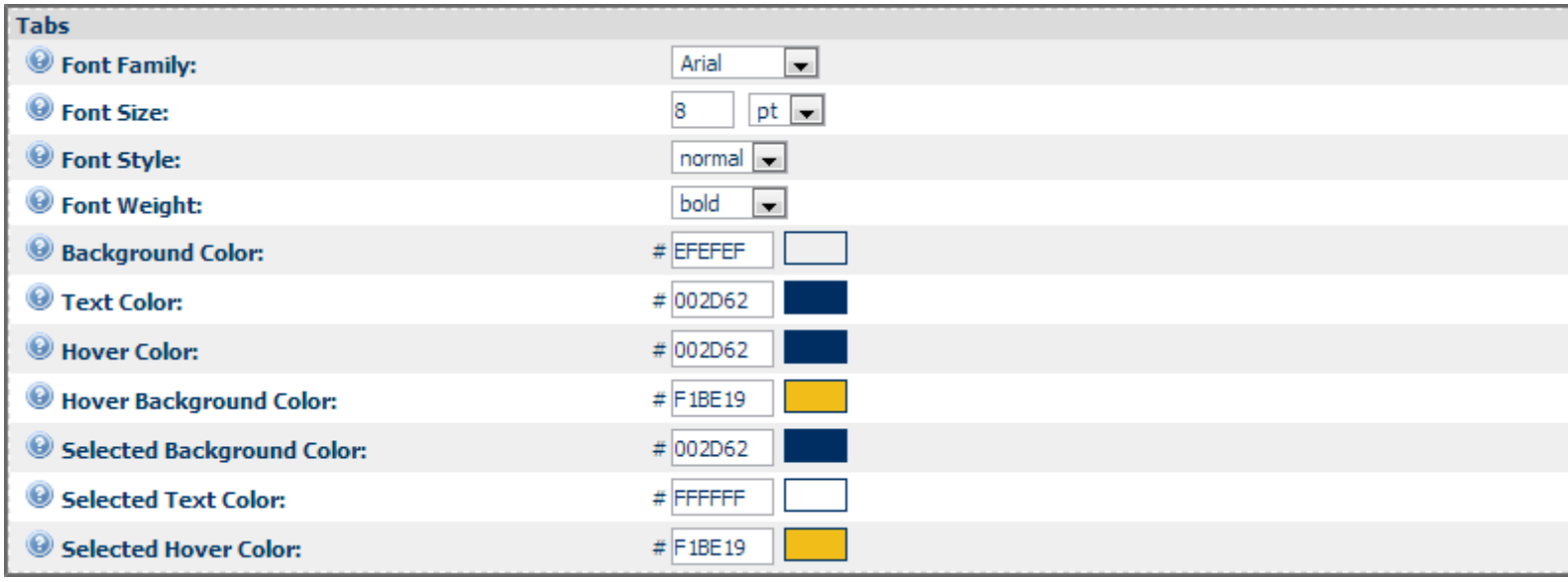
Commerce pages typically have multiple sections. Headers mark the start of a section. **Example:** on the Branding page, a few headers are: Top and Bottom Bars, Headers, Table Row Preferences, and Buttons.



Setting	Description
Background Color	Specify here the background color of the headers that mark the start of a section. You can see a preview of the color for the value entered.
Text Color	Specify here the color of the text that will be displayed on the headers that mark the start of a section. You can see a preview of the color for the value entered.
Hover Color	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.

Tabs

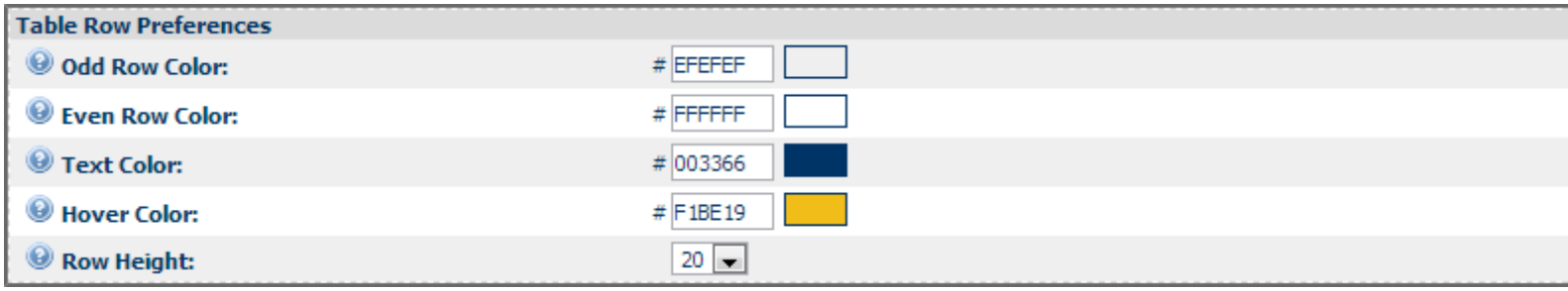
Tabs are used to provide structure to a web page by logically dividing the material in a page.



Setting	Description
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed.

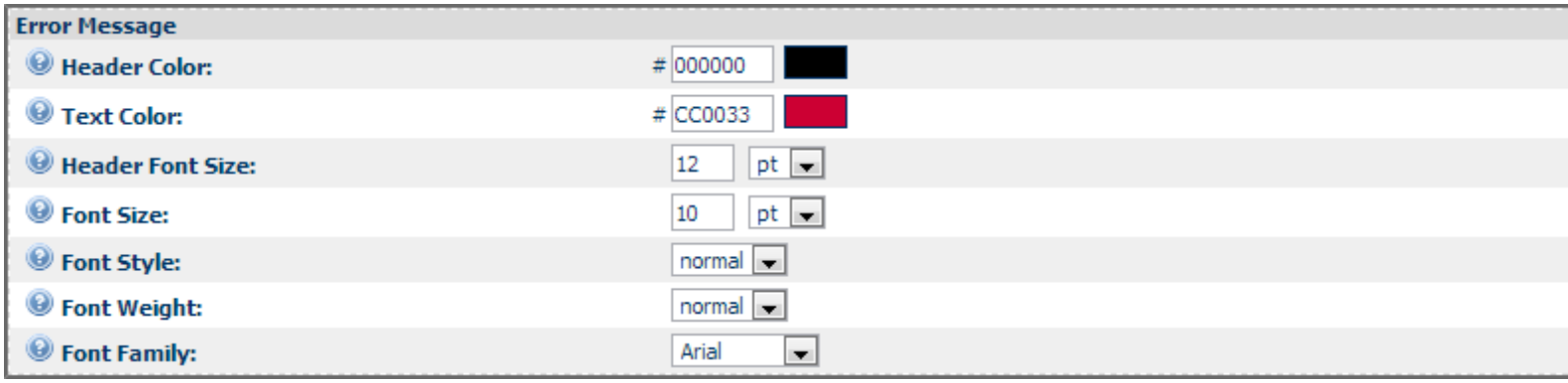
Table Row Preferences

Much of the data that displays on commerce pages is displayed in tables. You can alternate row colors to make tables more reader-friendly. You can also distinguish table text with a custom color.



Setting	Description
Odd Row Color	Specify here the background color of the odd-numbered rows of the table. You can see a preview of the color for the value entered.
Even Row Color	Specify here the background color of the even-numbered rows of the table. You can see a preview of the color for the value entered.
Text Color	Specify here the color of the text that will be displayed on the bars. You can see a preview of the color for the value entered.
Hover Color	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the color for the value entered.
Row Height	Using the drop-down list provided, select the height of the rows of the table.

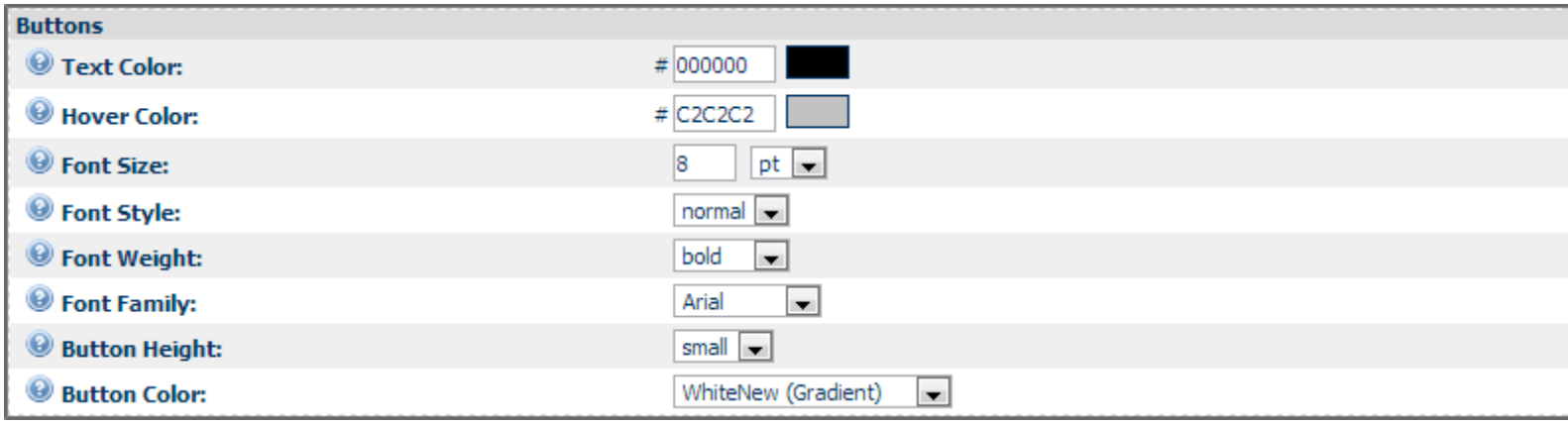
Error Message



Setting	Description
Header Color	Specify here the color of the error message header.
Text Color	Specify here the color of the error message text that will be displayed. You can see a preview of the color for the value entered.
Header Font Size	Using the drop-down list provided, select the font size of the error message header.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed in the error message.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed in the error message.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed in the error message.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed in the error message. The default font is Tahoma.

Buttons

Buttons enable users to perform operations. You can customize the background, text color, and sizes of all the buttons on your site.



Setting	Description
Text Color	Specify here the color of the text that will be displayed on the buttons. You can see a preview of the color for the value entered.
Hover	Specify here the hover color - that is the color in which the text is displayed when you bring your mouse pointer over it. You can see a preview of the

Setting	Description
Color	color for the value entered.
Font Size	Using the drop-down list provided, select the font size of the text that will be displayed on the buttons.
Font Style	Using the drop-down list provided, select the font style of the text that will be displayed on the buttons.
Font Weight	Using the drop-down list provided, select the font weight of the text (normal, bold, etc.) that will be displayed on the buttons.
Font Family	Using the drop-down list provided, select the font family of the text that will be displayed on the buttons. The default font is Tahoma.
Button Height	Using the drop-down list provided, select the height of the button.
Button Color	Using the drop-down list provided, select the color of the buttons.

ADMINISTRATION

Customizing your Stylesheet

1. Specify the fonts and color palette of your choice.
2. Do one of the following:
   - **To preview settings:** Click Preview. The refreshed view of the Site Branding page displays the new settings. To adjust the settings, make changes and then click Preview. When the page appears the way you want it to look, click Accept.
   - **To accept settings:** Click Accept.



Branding changes take effect immediately after clicking Accept. If the changes do not appear on a page, it is likely the browser is displaying a cached copy of the page. When this happens, click the browser's Refresh button to update the page.

When using top navigation for your site, the sidebar background color should be set to white (FFFFFF) and the text should be a dark, high-contrast color.

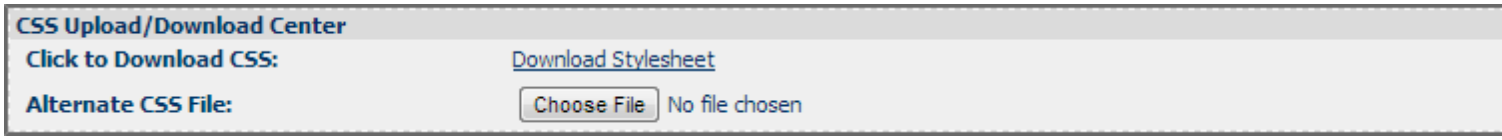


Only websafe colors can be selected from the color palette popup. RGB color codes display in the text-box next to the color preview box. *For non-websafe colors, you can enter RGB color codes and the corresponding color will be shown in the color preview box.*

Downloading a CSS File

This file is useful for developing XSL template files as it enables you to view the classes you can use in XSL.

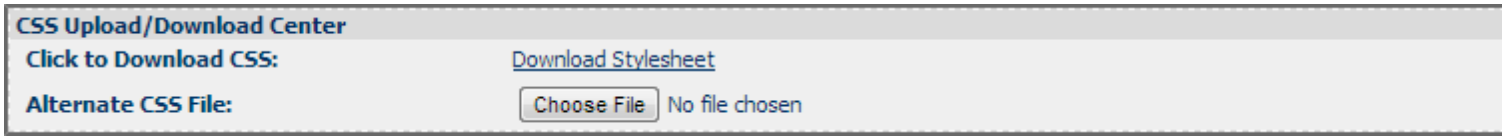
1. Click the **Download Stylesheet** link.



2. Save the downloaded CSS file, modify it, and then update it as an alternative CSS file.

Uploading a CSS File

1. Click **Browse** in the **Alternative CSS Information**. *This displays the File Upload dialog box.*

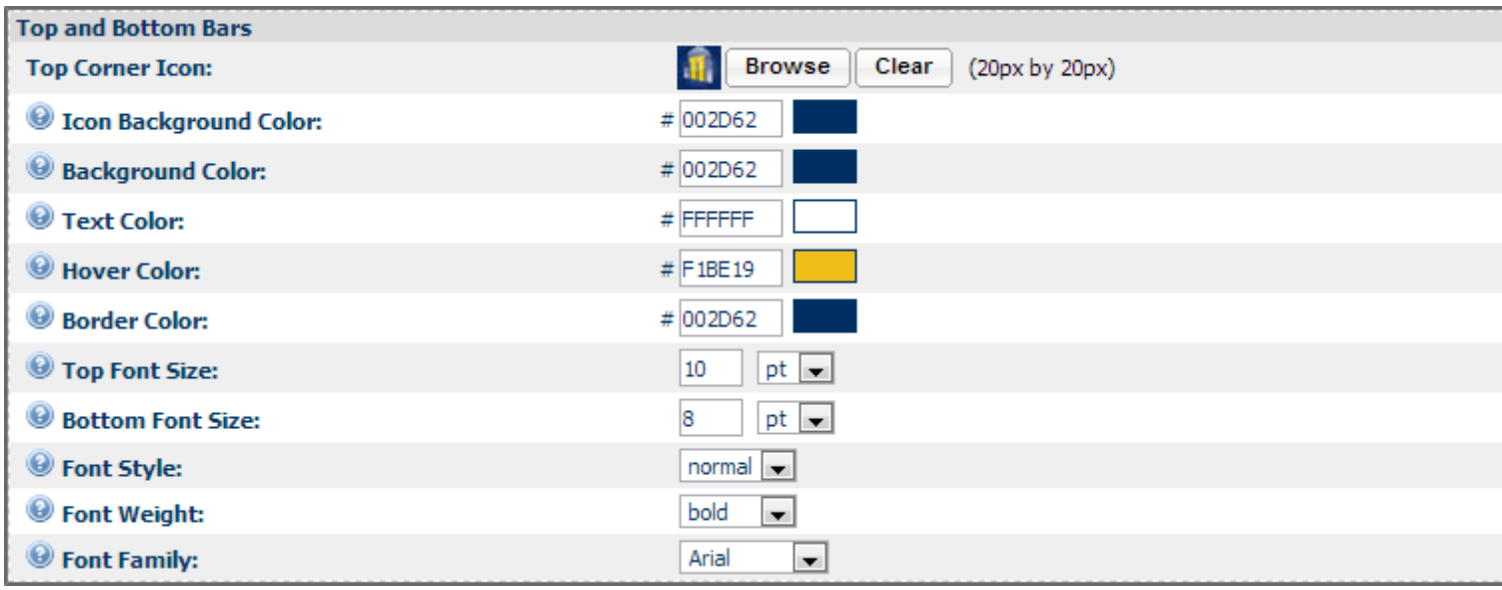


2. Browse for, select the required file, and click **Open**. This displays the selected file name for the **Alternative CSS Information** field.

3. Click **Accept** on the **Stylesheet Manager** page.

Specifying Icons for Top & Bottom Bars

1. Click **Browse** within the **Top Corner Icon** bar to search for and select a new logo. *This will open a new window, File Manager-Browse.*



2. Select the folder from which you desire to import the new logo, using the **Folder** drop-down list. *Selecting a folder name displays the logo images present in the folder, along with a Preview, Description, and the date and time the logo was imported to the folder.*

3. Click Select the entry for the desired logo. *This closes the **File Manager - Browse** window, and displays the name of the selected image in the **Home Page Set Up**.*

4. Click **Clear** to remove the selected icon. *The maximum size allowed for the icon is 20px by 20px.*

NOTES



By default, the sidebar contains the site's navigation links. Alternatively, navigation links can also be displayed across the top of the page. If you use sidebar navigation, then you can customize the sidebar's background color, text color and background image. *The width of a sidebar image should be a maximum of 150 pixels.*

It is recommended that you select one font to use across your entire site; it lends consistency to your site. *The default font is Tahoma.*

The drop-down list displays websafe colors.



Information regarding design principles:
-Smashing Magazine
-UX Design
-Yahoo UX
-Lynda.com
-Eric A. Meyer Blog
-A List Apart
-W3C Markup Validation Service
-QuirksMode.org

RELATED TOPICS

EDITING STYLESHEETS

OVERVIEW

You can use the following options to modify the style sheet being used for the Navigation menus:

- CSS Upload/Download Center
- Tabs (Admin, Commerce, Subheader, and Subfooter) to modify the look and feel of:
    - *Navigation bar*
    - *Delimiter (Top Navigation Only for Admin and Commerce)*
    - *Tabs (Top Navigation Only for Admin and Commerce)*
    - *Table Border (Top Navigation Only for Admin and Commerce)*

Using the CSS Upload/Download Center, you can either download or upload the CSS file:

- Click Download Stylesheet to download the current CSS. This file is useful for developing XSL template files because it enables you to view the classes you can use in XSL. For more information on developing XSL templates, see XSL Templates.
- Click Browse and follow the instructions shown in the File Upload dialog box to download your own CSS file in the Alternative CSS File field. Click Accept.

ADMINISTRATION

Customizing the Navigation Bar

Properties	Description
Background Image *(option not available for specifying if Subheader or Subfooter tabs are selected)*	Click Browse and follow the instructions shown in the File Upload dialog box to download your own CSS file in the Alternative CSS File field. Click Accept.
Background Color *(option not available for specifying if Subheader or Subfooter tabs are selected)*	Specify the background color using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Text Color	Specify the text color using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Hover Color	Specify the hover color using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Navigation Bar Spacer *(option not available for specifying if Subheader or Subfooter tabs are selected)*	Select the color for the navigation bar spacer using the drop-down list provided. Once decided upon the color, click Update to make the change permanent.
Font Family	Select the font-family using the drop-down list provided. Once decided upon the family, click Update to make the change permanent. The default font is Tahoma.

Properties	Description
Font Size	Select the font-size using the drop-down list provided. You can also specify the font size in either points or pixels. Once decided, click Update to make the change permanent.
Font Style	Select the font-style using the drop-down list provided. Once decided, click Update to make the change permanent.
Font Weight	Select the font-weight using the drop-down list provided. Once decided, click Update to make the change permanent.
Font Transform	Select the text-transform style using the drop-down list provided. Once decided, click Update to make the change permanent.

Customizing the Tabs (Top Navigation Only)

Properties	Description
Background Image *(option not available for specifying if Subheader or Subfooter tabs are selected)*	Click Browse and follow the instructions shown in the File Upload dialog box to download your own CSS file in the Alternative CSS File field. Click Accept.
Background Color *(option not available for specifying if Subheader or Subfooter tabs are selected)*	Specify the background color using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Text Color	Specify the text color using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Hover Color	Specify the hover color using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided upon the color, click Update to make the change permanent.
Navigation Bar Spacer *(option not available for specifying if Subheader or Subfooter tabs are selected)*	Select the color for the navigation bar spacer using the drop-down list provided. Once decided upon the color, click Update to make the change permanent.
Font Family	Select the font-family using the drop-down list provided. Once decided upon the family, click Update to make the change permanent. The default font is Tahoma.
Font Size	Select the font-size using the drop-down list provided. You can also specify the font size in either points or pixels. Once decided, click Update to make the change permanent.
Font Style	Select the font-style using the drop-down list provided. Once decided, click Update to make the change permanent.
Font Weight	Select the font-weight using the drop-down list provided. Once decided, click Update to make the change permanent.

Customizing the Delimiter (Top Navigation Only)

Properties	Sub-Options	Description
Delimiter Option		Select whether you want the delimiter to be an Icon or a Character using the radio-buttons provided. Once decided, click Update to make the change permanent.

Properties	Sub-Options	Description
Icon		Selecting the Delimiter option as an Icon displays the following options.
	Icon	Click Browse to open the File Manager. You can scroll through the images in the File Manager by clicking Next or a tab name. To select an image, click the image thumbprint. This displays the selected icon next to Browse. Click Update to make the change permanent.
	Width: px	Specify the Width of the icon in pixels.
	Height: px	Specify the Height of the icon in pixels.
Character		Selecting the Delimiter option as a Character displays the following option.
	Character	Select the delimiter character to be used for separating the various options using the drop-down list provided. Once decided, click Update to make the change permanent.

Customizing the Table Border (Top Navigation Only)

Properties	Description
Navigation Table Border Color (option not available for specifying if Subheader or Subfooter tabs are selected)	Specify the border color of the table being navigated using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided, click Update to make the change permanent.
Header Table Border Color (option available ONLY IF Subheader tab is selected)	Specify the border color of the header of the table being navigated using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided, click Update to make the change permanent.
Footer Table Border Color (option available ONLY IF Subfooter tab is selected)	Specify the border color of the footer of the table being navigated using the hexadecimal format. For a preview of the color, click Update. The color is displayed in the color box next to the field. Once decided, click Update to make the change permanent.

NOTES



Changes take effect immediately after clicking Update. If the changes do not appear on a page, it is likely the browser is displaying a cached copy of the page. When this happens, click the browser's Refresh button to update the page.

When top navigation is chosen as a template, the sidebar value should be set to white (FFFFFF).



The system's Stylesheet can be downloaded for viewing purposes only. It is not possible to modify the CSS and then upload it back into the system; however, it is possible to upload alternative stylesheet to



override the setting the system's stylesheet.



It is not possible to preview the CSS setting until you accept it.

Only websafe colors can be selected from the color palette popup. *RGB color codes display in the text-box next to the color preview box.* For non-websafe color, you can enter RGB color codes and the corresponding color will be shown in the color preview box.

RELATED TOPICS

HTML TEMPLATE EDITOR

OVERVIEW

The HTML Template Editor is available in several administration areas, including company information, site customization, and product administration. The HTML Template Editor supports a subset of HTML tags and JavaScript. It verifies HTML formatting before saving data and supports only those tags that won't compromise the security of your system. The editor can be launched from administration areas whenever you see an **Edit HTML** template button. The tool opens in a separate browser window and provides a text editor window for entering HTML-formatted content. Widgets within the editor enable you to quickly add images and create roll-overs. The HTML Editor also supports automatic entry of valid HTML tags.

Available Actions

Actions	Descriptions
Preview	Enables you to view the HTML as it would appear to buyers, channel agents and sales agents. *A good way to test how your HTML will display.*
Check and Save	Enables you to save your HTML content. The HTML Editor verifies your HTML automatically before saving it. When errors exist, like unsupported tags, an error message displays and the content is not saved. *Errors must be fixed before content can be saved.*
Check	Verifies that the content is error-free. Error messages display when problems exist.
Cancel	Exits out of the HTML Template Editor. No information is saved.

ADMINISTRATION

Copying HTML

1. Copy content from the third party editor to a text editor, such as Notepad++.
2. Remove any visible characters that are not supported.
3. Copy and paste the content from the text editor into the HTML Editor.



Some HTML editors create invisible characters that the HTML Editor tool cannot handle.

Creating Rollovers

Roll-overs are dynamic layered images, where a top image displays by default and a bottom image displays when a user's mouse hovers over the image area. Roll-overs work best when the two images are the same size.

1. Click **Rollover** on the HTML Editor page.
2. In the Create Rollover dialog, enter two images individually.
3. To associate an image, click **Browse**. *This opens a separate window containing the contents of the File Manager.*

4. In the Image Browser window, select an image by clicking the image thumbprint.
5. Enter a **Name** for the image. *This name does not appear with the images, it is just used in the JavaScript created for the rollover.*
6. Enter a URL for the image if you want to be a hyperlink.
7. Click **OK**. *The JavaScript function calls for the rollover appears at the current cursor location in the text area. For security reasons, the actual JavaScript function does not display.*

Special Characters

Certain characters are reserved by HTML to refer to the markup language. To use a reserved character as content, specify it with its entity reference. The Special Characters section of the HTML Editor enables you to bypass looking up entity references. You can quickly select a character, and automatically add its entity reference to your HTML content. **To enter a special character:**

1. Highlight the character from the drop-down list box in the **Special Characters** section.
2. Click **Add**. *The entity reference is added at the current cursor location in the text area.*

Valid Tag References

The subset of HTML tags supported in the HTML Template Editor is listed in the **Valid Tags** reference window. You can view this list, and automatically add tags from the list. **To use the valid tags reference:**

1. Click **Display** in the Valid Tags section.
2. To automatically enter a tag, click a **Tag Name**. *The tag is added at the current cursor location in the text area.*

NOTES



You can enter special characters by manually typing their entity references. For a complete list of entity references, refer to the World Wide Web Consortium.



Tags relating to forms and frames are not supported due to security concerns. *For flexibility, JavaScript is supported.* You can also enter tags manually by typing them into the text area.

Normal Image (image-off) refers to the image that displays anytime the mouse is not hovering over the image. **Highlighted Image** (image-on) refers to the image that displays anytime the mouse is hovering over the image.

Images can be selected from any folder in the Image Browser window. The images contained in the



[Default] folder appear when the window first opens. *To view images in a different folder, select it from the drop-down list provided.*



The rollover widget can only use images that exist in the File Manager utility.

RELATED TOPICS

HOME PAGE FLOWS

OVERVIEW

The system supports a set of home page flows for the purpose of defining what items are displayed on the catalog home page. When you select a flow, you choose whether parts or products, or a combination of both, should appear on the home page. Home page flows are used to decide how the system's home page looks, and has an impact on subsequent navigation (including catalog navigation). There are flows available for supporting a parts-only catalog, products-only catalog, and a combination catalog of both products and parts.

Example: Your company supports multiple product families, you could select the flow that displays a set of product families on the home page so users can begin product searches with product family selection. Or, if your company supports a single product family, the flow that displays product lines on the home page would be more suitable because it permits users to bypass the unnecessary step of product family selection.

Enabling a Home Page Flow

1. Select the flow description's corresponding radio button.
2. Click **Save**. *Template changes take effect immediately after clicking* ***Save****.*

ADMINISTRATION

Available Commerce Home Page Flows:

Multiple Equipment Segments

Template	Description
Template 1	This template has two sections: top and bottom. Your company description and any embedded HTML images, appear across the top of the page. Supported product families appear at the bottom.
Default	This template has three sections: top left, top right and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. Supported product families are listed at the bottom.

Parts Search & Multiple Equipment Segments

Template	Description
Template 1	This template has three sections: top, middle and bottom. Your company description, and any embedded HTML images, appears across the top of the page. The Parts Search template appears in the middle, and supported product families are listed at the bottom of the page.
Default	This template has four sections: top left, top right, middle and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. The Parts Search template appears in the middle of the page. Supported product families are listed at the bottom.



The home page must be set to **Custom** to see these options.

Many Product Lines within a Single Segment

Template	Description
Template 1	This template has three sections: top, middle and bottom. Your company description, and any embedded HTML images, appears across the top of the page. A commerce section, accelerating commerce document creation, appears in the middle, and the Attribute Search template appears at the bottom of the page.
Default	This template has four sections: top left, top right, middle and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. A commerce section, accelerating commerce document creation, appears in the middle, and the Attribute Search template appears at the bottom of the page.



The home page must be set to **Custom** to see these options.

Parts Search & Many Product Lines within a Single Segment

Template	Description
Template 1	This template has three sections: top, middle and bottom. Your company description, and any embedded HTML images, appears across the top of the page. The Parts Search template and a commerce section, accelerating commerce document creation, appears in the middle, and the Attribute Search template appears at the bottom of the page.
Default	This template has four sections: top left, top right, middle and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. The Parts Search template and a commerce section, accelerating commerce document creation, appears in the middle, and the Attribute Search template appears at the bottom of the page.



The home page must be set to **Custom** to see these options.

A Few Product Lines within a Single Segment

Template	Description
Template 1	This template has two sections: top and bottom. Your company description, and any embedded HTML images, appears across the top of the page. The Product Line List template displays at the bottom of the page.
Default	This template has three sections: top left, top right and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. The Product Line List template displays at the bottom of the page.



The home page must be set to **Custom** to see these options.

Parts Search & a Few Product Lines within a Single Segment

Template	Description
Template 1	This template has three sections: top, middle and bottom. Your company description, and any embedded HTML images, appears across the top of the page. The Parts Search template displays in the middle, and the Product Line List template displays at the bottom of the page.
Default	This template has four sections: top left, top right, middle and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. The Parts Search template displays in the middle, and the Product Line List template displays at the bottom of the page.



The home page must be set to **Custom** to see these options.

Single Product Line

Template	Description
Template 1	This template has two sections: top and bottom. Your company description, and any embedded HTML images, appears across the top of the page. The Product Line Profile template displays at the bottom of the page.
Default	This template has three sections: top left, top right and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. The Product Line Profile template displays at the bottom of the page.



The home page must be set to **Custom** to see these options.

Parts Search & A Single Product Line

Template	Description
Template 1	This template has three sections: top, middle and bottom. Your company description, and any embedded HTML images, appears across the top of the page. The Parts Search template appears in the middle, and the Product Line Profile template displays at the bottom of the page.
Default	This template has four sections: top left, top right, middle and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. The Parts Search template appears in the middle, and the Product Line Profile template displays at the bottom of the page.



The home page must be set to **Custom** to see these options.

Parts Search

Template	Description
Template 1	This template has two sections: top and bottom. Your company description, and any embedded HTML images, appears across the top of the page. The Parts Search template appears at the bottom.
Default	This template has three sections: top left, top right and bottom. Your company logo displays top left. Your company description, and any embedded HTML images, appears top right. The Parts Search template appears at the bottom.



The home page must be set to **Custom** to see these options.

hp_flow10

Template	Description
Default	It's a blank page, you can design your own home page, including the part search and supported product families as you want.

NOTES



Flow template descriptions are provided under Page Templates and User Interface > Page Templates.

To preview flow templates, use the following path: Page Templates and User Interface > Page Templates > click a Description > displays the Available Templates for the selected description.

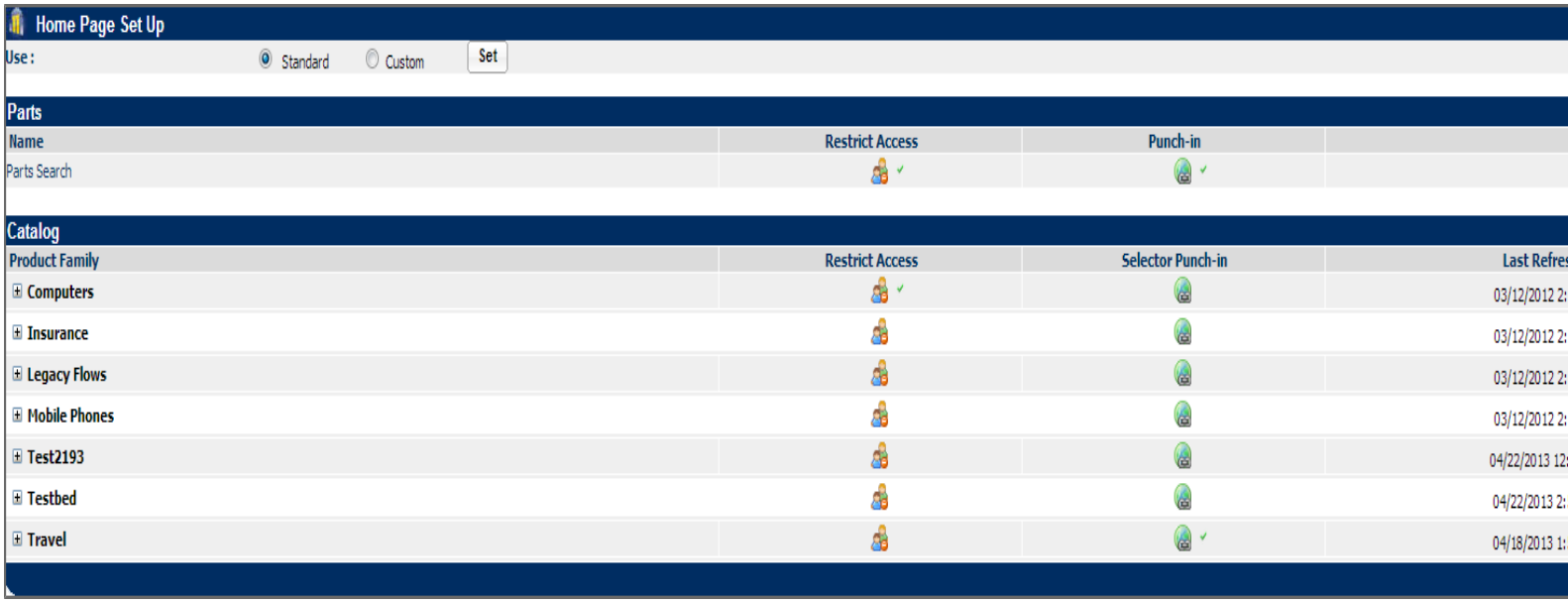
RELATED TOPICS

HOME PAGE

OVERVIEW

You can customize the home page of the web site using HTML and JavaScript or XSL. You, the fullaccessuser, can set up the web page in any way you choose in conjunction with the Home Page Flows that are made available in the Page Templates and User Interface section. Typically, the home page serves as a portal to product selectors, model configurators, parts search, company information, general product documentation, and/or external links.

Navigation: Admin Homepage > Application Branding & Home Page > Home Page > Standard OR Custom



ADMINISTRATION

Types of Home Pages:

Standard Home Page- XSL

You can use a default XSL Home Page to display your products and parts. You can choose what is displayed on the home page based on the user.

Available Actions

Action	Description
Check	Check for errors.
Run	See console output for the print statements in the script.
Save	Save changes made.
Save and Close	Save content and close window.
Close	Close the window without saving the changes made.
Always Deny User Access	Denies access to product families, product lines, or parts searches for all users.



These actions can be performed once you have entered/modified the script.

Parts Section

Clicking the **User Access** icon displays a new window for selecting attributes to define a new rule. **The available attributes are:**

- _user_company
- _user_currency
- _user_email
- _user_login
- _user_groups
- _user_language
- _user_number_format
- _user_type
- _company_type
- _user_number_format_str
- user_type_str
- _company_type_str

Selecting the desired attributes and clicking **Rule Editor** refreshes the window and displays the **User Access Rule Editor** page.

Rule Editor Page: What You Can Do

- Select **Simple Conditions** and specify values for the selected attributes to create a simple conditions rule.
- Select **Advanced Function** and click **Define Function** to define the script to create an advanced rule. *Clicking **Define Function** opens a new window, Function Editor, to enter a script for an advanced rule.*
    - Clicking a **Variable Name** displays the name in the **Function Script** area where you can enter or modify the script.
    - Clicking an icon for an **Operator** displays it in the **Function Script** area.
    - Clicking the **Editor Help** displays the help for the Function Editor Information in a new window. The help displays all topics for the **Functions**, **Statements**, and **Operators**.

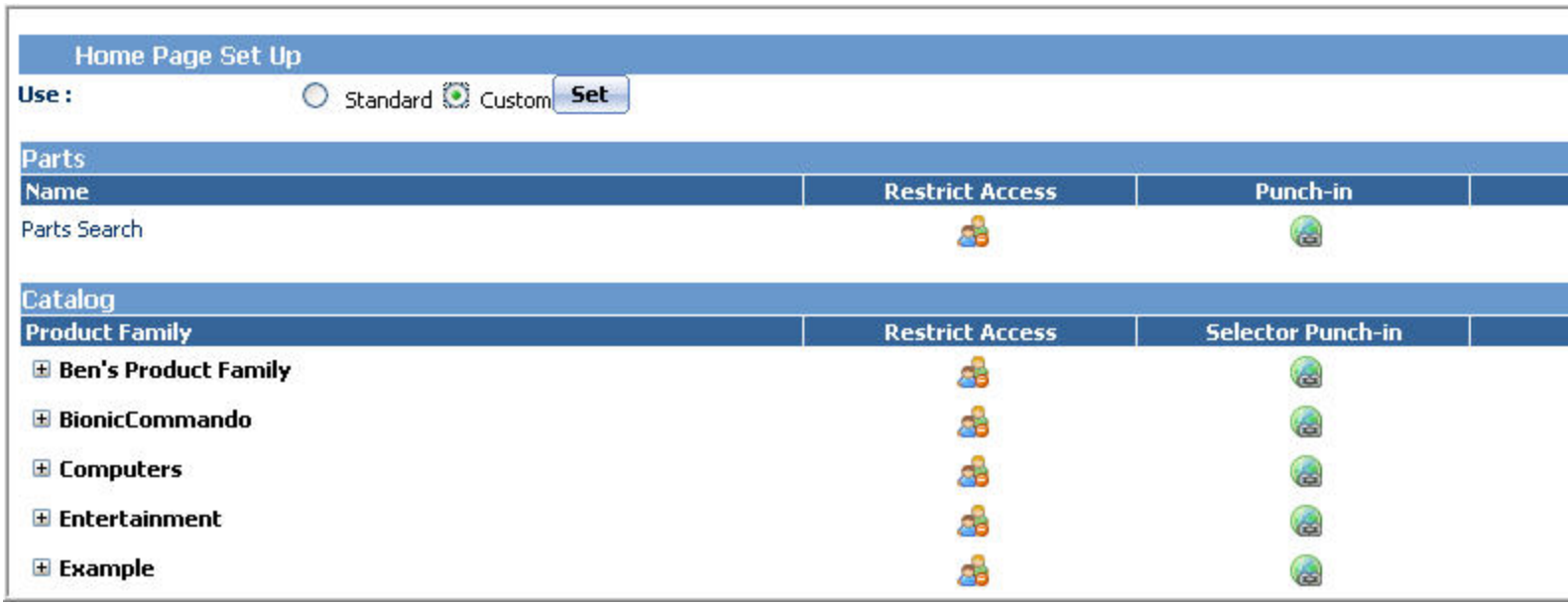
- Select the desired function from the **Functions** drop-down list. Selecting a function name displays a brief definition and the syntax for the selected function. Clicking Add displays the selected function in the **Function Script** area.
- Clicking the **Punch-in** icon displays a new window - **Part Detail Punch-ins List** - to create a new custom punch-in.

Catalog Section

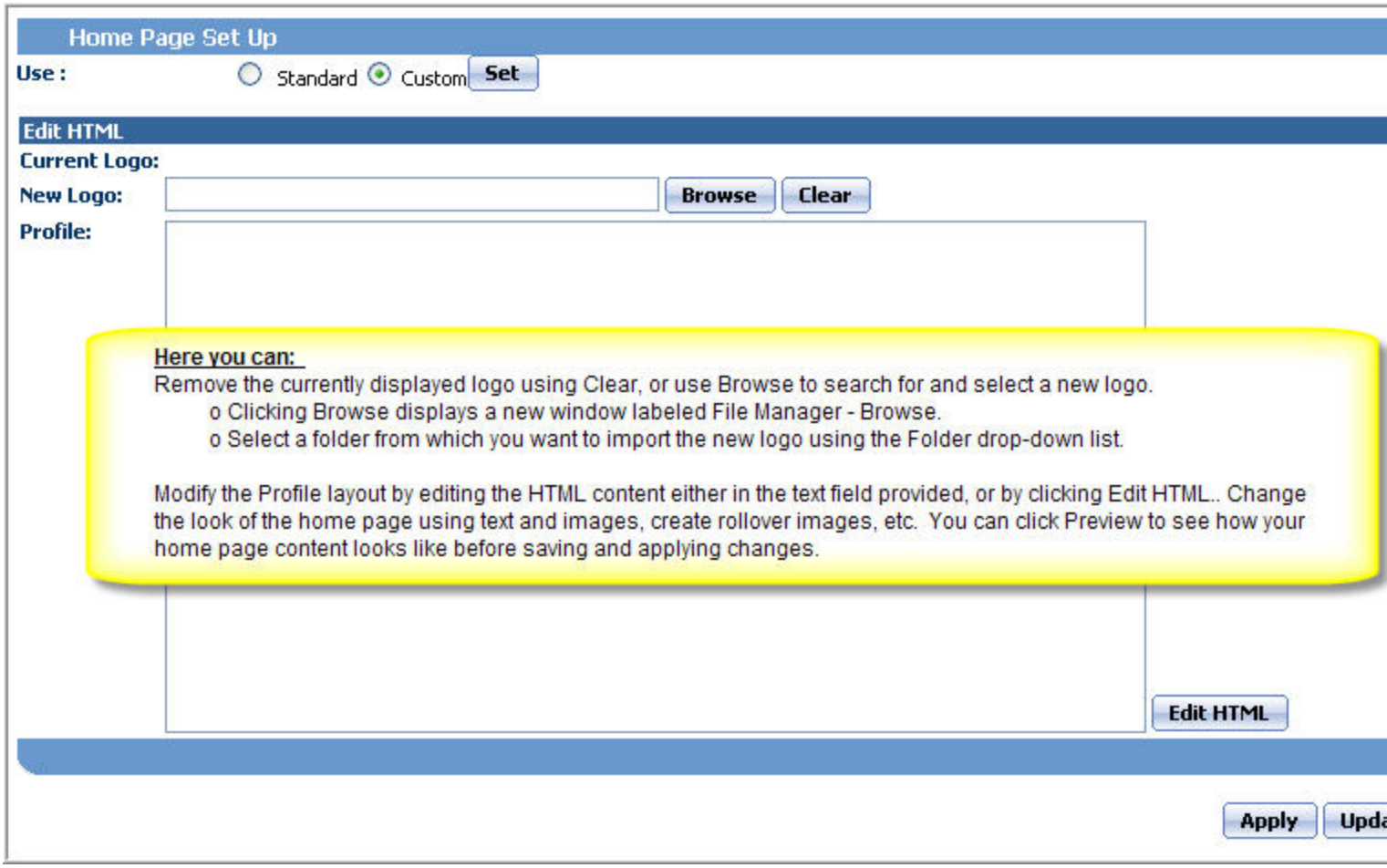
- Selecting a **Product Family** name displays the list of **Product Lines** for the same.
- Clicking the **User Access** icon displays a new window for selecting attributes to define a new rule. Selecting the desired attributes and clicking **Rule Editor** refreshes the window and displays the **User Access Rule Editor** page. Using this page you can:
  - Select **Simple Conditions** and specify values for the selected attributes to create a simple conditions rule.
  - Select **Advanced Function** and click **Define Function** to define the script to create an advanced rule. Clicking **Define Function** opens a new window - **Function Editor** - to enter a script for an advanced rule.
    - *Clicking a **Variable Name** displays the name in the **Function Script** area where you can enter or modify the script.*
    - *Clicking an icon for an **Operator** displays it in the **Function Script** area.*
    - *Clicking the **Editor Help** displays the help for the Function Editor Information in a new window. The help displays all topics for the **Functions**, **Statements**, and **Operators**.*
    - *Select the desired function from the **Functions** drop-down list. Selecting a function name displays a brief definition and the syntax for the selected function. Clicking **Add** displays the selected function in the **Function Script** area.*
- Clicking the **Selector Punch-in** icon displays a new window, **Selector Punch-in List**, allowing you to create a new custom punch-in. If you select the **Selector Punch-in icon for a Product Family** and click **New**. Clicking the Selector Punch-in icon for a Product Line and clicking New refreshes the Selector Punch-ins List page and displays options to create a new Selector Punch-ins List. *You will NOT be able to add a new punch-in if there are no associated Search Groups available.*
- Clicking the **Last Refreshed On** icon shows the date on which the catalog data was last refreshed for a product family. Click this icon to create/update/delete default punch-in links corresponding to the latest catalog data.
- Clicking the **Model Punch-in** icon displays a new window - **Model Punch-ins List**.
- Click **Deployment Center** to deploy the latest home page data.

To launch the **Admin XSL Template** page, click on **Define XSL Template**. You can display default templates used. You can also download the default template, modify it, and then upload it.

Custom Home Page- HTML



Customize your homepage using HTML



Home Page changes take effect immediately after clicking ‘Apply’ or ‘Update’.

Best Practices

- Use an external editor, such as Dreamweaver or Notepad ++, to create your HTML content; then, copy/paste your file directly into the text area.
- Keep your user in mind. Ensure that they can easily navigate within the page.
- When creating your HTML Home Page, keep in mind that your Header & Footer HTML will also be displayed on your Homepage. For more information, see Header & Footer.

RELATED TOPICS

HEADER & FOOTER

OVERVIEW

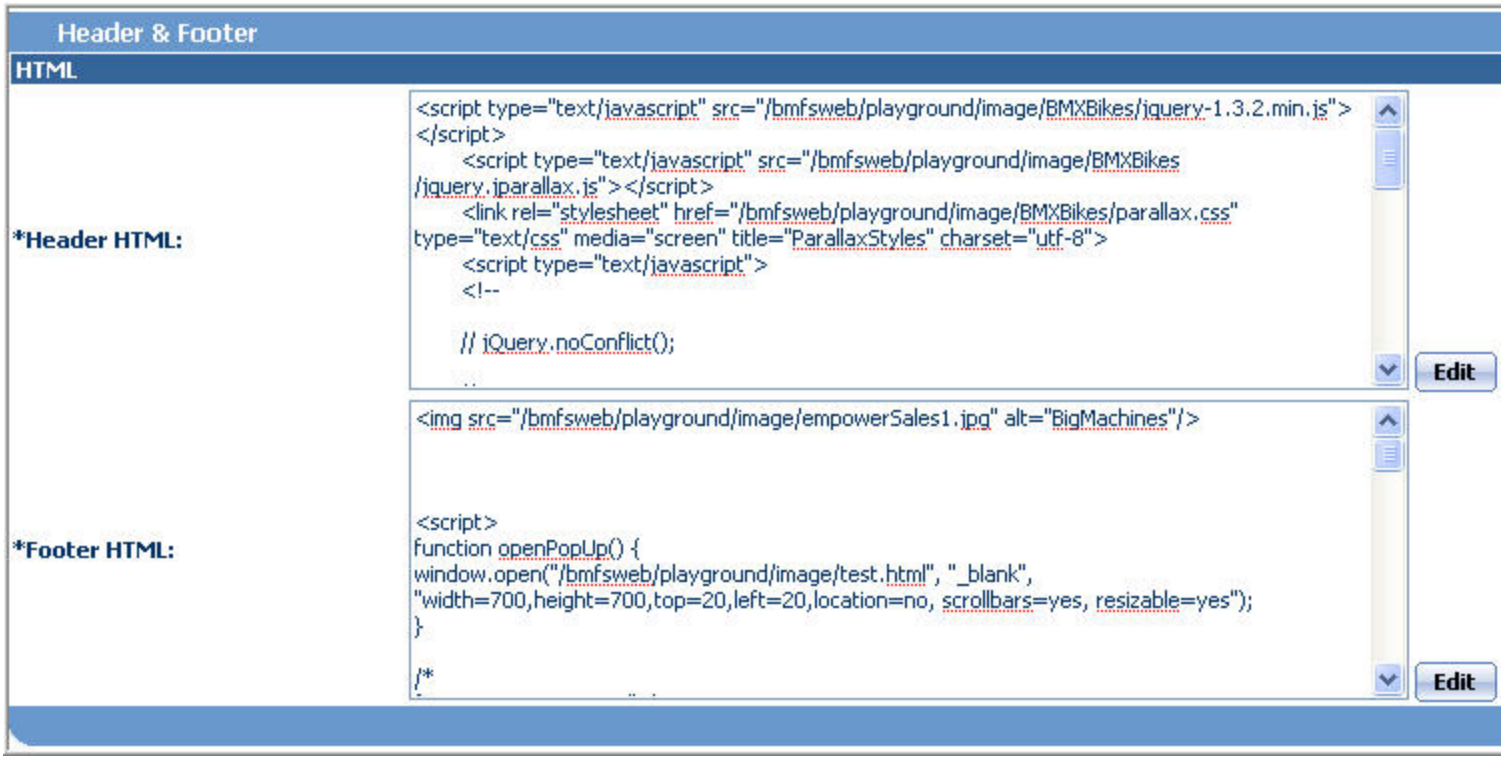
You can customize an HTML Header and Footer that will be visible across all pages of the site. Sometimes, you can use this to your advantage by placing Javascript code that you will need to access from other pages of the site in the header or footer.

Navigation: Admin Home Page > Styles and Templates > Header and Footer

ADMINISTRATION

Creating a Header or Footer

1. Do one of the following:
   - **To create a header:** Click the **Edit** button for the Header HTML field.
   - **To create a footer:** Click the **Edit** button for the Footer HTML field
2. Make changes to the HTML content for the header (or footer) using text and images.
3. You can either directly enter the HTML into the boxes, or click on the respective edit button to bring up the HTML editor:



4. Click Preview to see how the new page will look, click Accept to save the change or click Cancel to go back to the Administration Home page.

Best Practices

- In addition to providing consistent branding throughout your site, the Header and Footer can also be used to your advantage by placing HTML to create: menus, navigation bars, drop-down menus, rollover effects, floating boxes, or sidebars. *All of this can be done through CSS and does not require any JavaScript.*
- You can use an external editor such as Dreamweaver or Notepad ++ to create your HTML content, and then copy/paste it directly into the text area. *Using an external editor assists with syntax formatting and allows you to save backup copies of your work.*
- If you include images in your Header/Footer, upload the image file to the File Manager. *Avoid using external images.* For more information, see File Manager.

NOTES



To view a header or footer before saving it, click the Preview button. *Preview buttons are available in the HTML Editor and on the Site Header & Footer page.*



Custom headers and footers display on all commerce system pages, including pop-up windows.

Header and footer changes take effect immediately after clicking Accept.



If you do include JavaScript for other areas of the site in the header and footer, make sure that you have designed it carefully so it will not interfere with performance for pages other than the intended.

RELATED TOPICS

NAVIGATION MENUS

OVERVIEW

The navigation menu is a shortcut menu that lets you access desired areas using the side/top navigation menu. What appears in these menus can be customized based upon where the page you, or the user, happen to be working on: Admin, Commerce, Quick Registration, Subheader and Subfooter.

There are default menus for each user type, but you can customize these by adding custom links or hiding default links. Navigation menus can be customized with the links you want to display in the sidebar or top navigation bar. You can add, reorder, hide, or delete links. External links can be added to default link sets, and you can designate a unique link set for each user type.

Navigation Menu Descriptions

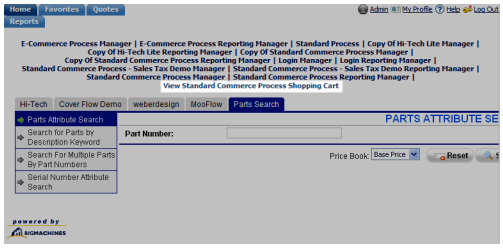
Navigation Menus are always managed in a pending state. In order for navigation menu changes to become active, they must be initialized on the system. To determine which user type has access to which menu, see the access rights tables for the various menus provided.

Navigation Menu	Description
Admin	The Admin menu contains links pertaining to administration functions in the system. In order for users to reach the administration menu, the Subheader menu must contain the default Admin link. The Admin link cannot be hidden from full access fullaccessusers in the FullAccessWithESales company type.
Commerce	The Commerce menu contains links to Commerce Managers, Commerce List, and the commerce system's home page where product searches can be conducted. The commerce area of the site automatically appears when a user logs in to the BigMachines system. When a user is in the administration area of the site, they can return to the commerce and catalog areas by clicking the Home Page link in the Subheader menu.
Quick Registration	The Quick Registration menu is displayed before a user logs in to the system. It is also displayed during the quick registration process.
Subheader	The Subheader menu enables users to toggle between the Admin and Commerce menus. This menu is always active when a user is logged in to the system, and by default it contains links to online help and the log out process.
Subfooter	The Subfooter menu refers to the links that are available at the end of a page.
Sidebar/Top Navigation	Display your favorite menu options vertically in the sidebar or horizontally on the top navigation bar. *Click Set to make the required change.* **Side Navigation Template:** all menus, except Subheader, appear in the side navigation bar. The Subheader menu always appears horizontally across the top of the page, either as a set of links (side navigation template) or a set of tabbed links (top navigation template). **Top Navigation Template:** all menus appear horizontally across the top of the page.

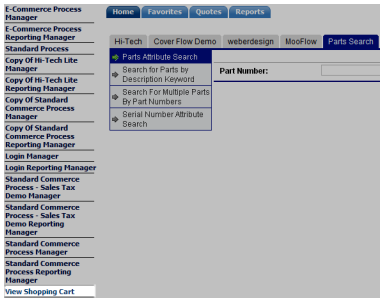
ADMINISTRATION

Setting the Navigation

1. Navigate from: Admin Home Page > General > Navigation Menus.
2. Determine where the navigation menu will be placed:
    - Top Navigation: Navigation menu will appear above the administration and user UI.



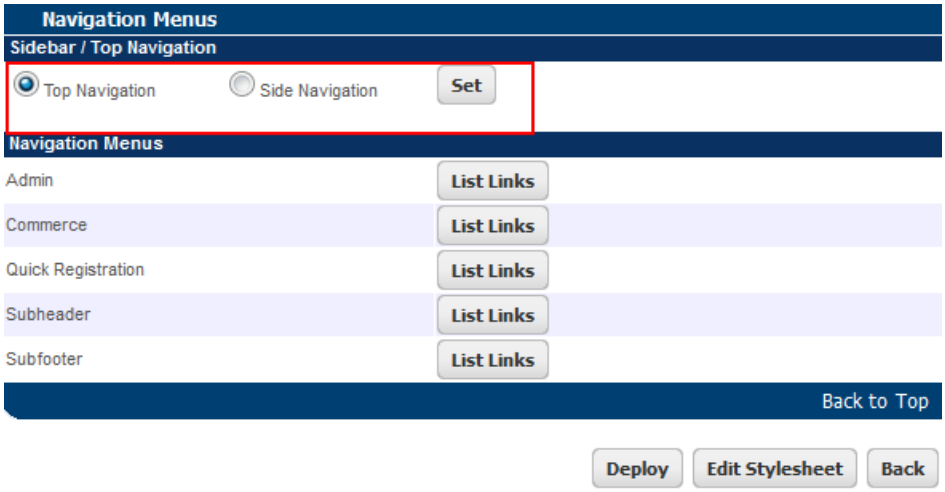
    - Side Navigation: Navigation menus will appear to the left of the administration and user UI.





All menus, except Subheader, appear in the side navigation bar. The Subheader menu always appears horizontally across the top of the page, either as a set of links (side navigation template) or a set of tabbed links (top navigation template).

3. Click **Set**.



Managing Navigation Menus

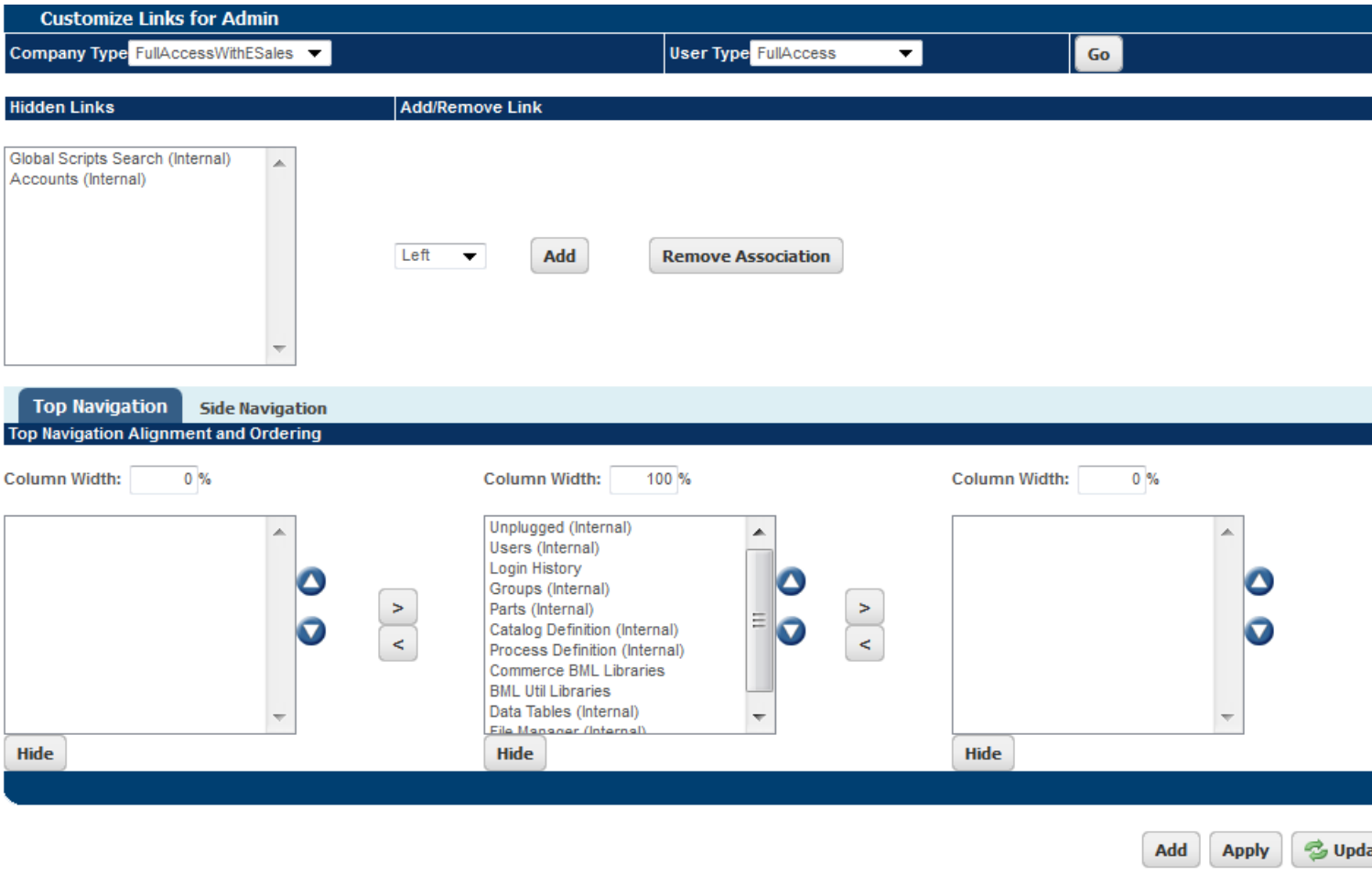
1. Navigate from: Admin Home Page > General > Navigation Menus.
2. Click **List Links** next to the menu you'd like to edit.
3. Click one of the following:
    - Internal Tab
    - User Defined Tab
    - Customize Menus

Customizing Menu Buttons

Use this area to set the company/user type that can access this menu. *You will also hide or show links here.*

1. Click **Customize Menus** on the Internal Tab or User Defined tab.
2. Choose a **Company Type** and **User Type** and click **Go**.
3. Do one of the following:
    - **To hide a default link:** Select the navigation item you want to hide from desired section and click **Hide**. Click **Update**.
    - **To add a default link:** Under the Hide Links section, select a link and click **Add**. Click **Update**.
    - **To remove a default link:** Under the Hide Links section, select a link and click **Remove Association**. Click **Update**.

4. Modify which columns are displayed in the **Top Navigation** and **Side Navigation** tabs:
    - Select the required tab.
    - Determine the column width. The sum of the column widths cannot exceed 100%.
    - Select the field that needs to be moved up or down in the column and use the upward and downward key combination to make the desired changes.
    - Select the field that needs to be moved to the other column and use the move to and move from key combination to make the desired changes.
    - To hide a displayed field, select it, and click **Hide**.

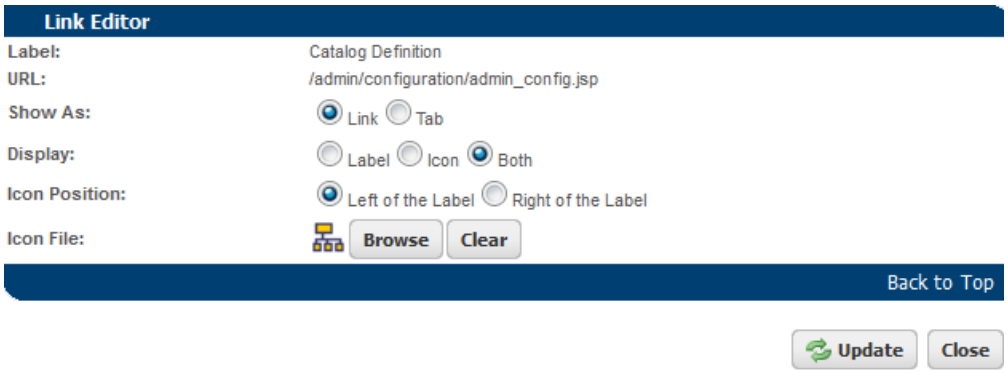


5. Click **Add** to add a custom link to the menu, if available.
6. Click **Update** after doing necessary changes
7. Click **Back** and this will take you to 'Navigation Menu' page. Click **Deploy** to deploy.

Editing Menu Links: Internal or User-Defined Tab

The Internal tab contains all the navigation items that are built-in.

1. Click **List Links** next to one of the navigation menus.
2. Click on the **Internal or User-Defined** tab.
3. Click **Edit** next to a corresponding link. *When you click **Edit** against an item, the corresponding 'Link Editor' page is displayed.*



4. Manage the following settings:
   - **Show As:** You can either show the item as a link or tab.
   - **Display:** Choose how to display the item, if you chose *Show As: Link*.
   - **Icon Position:** If you selected **Both**, you will need to decide if the icon should display to the left or right of the label.
   - **Icon File:**Click **Browse** to select an image to display next to the icon. *This is if you chose to display the link as an icon.*
5. Click **Update** to save your changes and return to the Navigation Items List for the menu you are working on.

Adding a Custom Link: User-Defined Tab

1. Click **List Links** next to one of the navigation menus.
2. Click the **User-Defined** tab.
3. Click **Add** to add a navigation item.
4. Enter the Label and the URL.
5. Click **Add**.

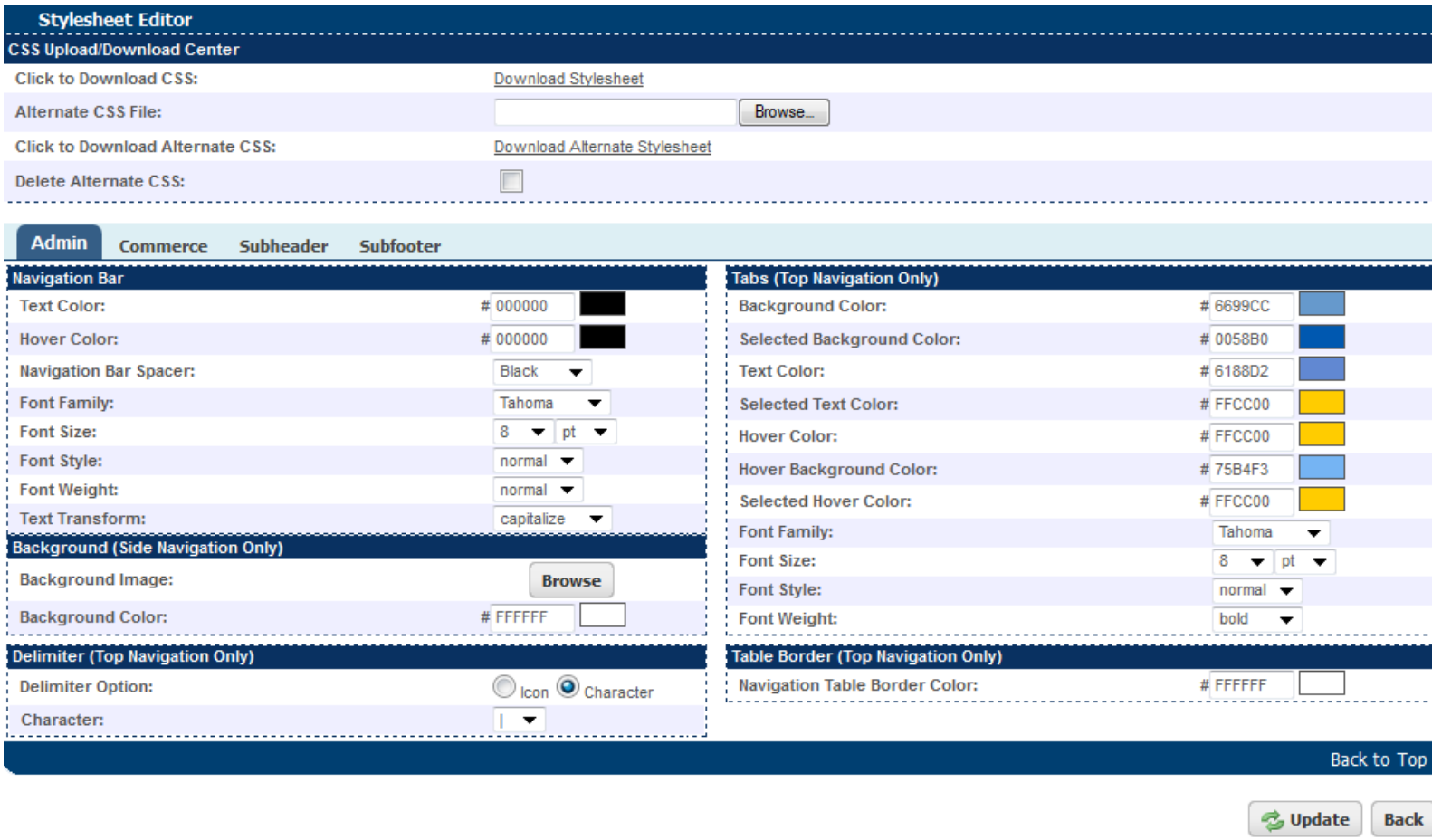


To delete user-defined links, select the links to be deleted and click Delete.

Translations

1. Click **Translations** to open a dialog where you can enter text language translations for supported languages.
2. Click **Save** to save translations in the system and continue entering translations.
3. Click **Save and Close** to save entered translations and close dialog. Click **Close** to close dialog without saving.

Editing the Stylesheet



Deploying the Navigation Menus

1. Navigate from: Admin Home Page > General > Navigation Menus.
2. Make edits to any navigation menus and their links.

3. Click **Deploy** on the Navigation Menus page.



All changes must be deployed in order to take effect.

NOTES



When using top navigation for your site, the sidebar background color should be set to white (FFFFFF) and the text should be a dark, high-contrast color.

After updating links, remember to deploy the navigation menu on the Navigation Menu List page.



Through the use of custom links, user types cannot be granted access rights to areas of the commerce system that they do not have default access to.

Customization of menu links can occur in: Admin, Commerce, Quick Registration, Subfooters, and Sub-headers.



When a link is deleted from the system, it is also removed from all menus.

User types cannot be granted access rights to additional areas of the commerce system through the use of links.

RELATED TOPICS

PAGE TEMPLATES

OVERVIEW

Templates offer different layouts for displaying catalog pages. The system supports templates for most catalog pages, enabling you to pick the layouts that work best for your system. Parts-specific catalog pages also support XSL page templates. For those pages, you can customize page display using XSL Stylesheets.

Page Templates control how data is displayed, not what data is displayed. **Example:** there are seven template options available for the Product Line List page, one default template and six alternative templates. The default template displays product line search results in a compact row format. The other templates offer alternative formats for the same data. So, if you prefer product line search results to display in column format, you could choose template 5 which offers column-based search results.

Editing Text

1. Do one of the following:
    - **To edit text:** In the **Translated Text** area, enter the words you want to use in place of the default system text. Click the **Override** checkbox.
    - **To revert to default text:** Disable the **Override** option.
2. Click **Update** to save the changes made.
3. Return to the **Centralized Naming and Title Editing** page.



Text changes take effect immediately after clicking the **Update** button.

Words and phrases are always sorted by the default system text, never by edited text phrases.



Page help text and user data cannot be edited in the text administration area.

Selecting a Template

1. Enable desired radio buttons to select a template.
2. Click the **Click to Preview** link to view the preview images of a template.
3. Click **Save**. *Template changes take effect immediately after clicking Save.*

ADMINISTRATION

For each catalog page, you can select one template for the production environment and one template for the test environment. The test environment enables full access fullaccessusers to preview how templates display data before the template is moved into production. *The test environment is a useful way to preview how home page flows appear on the system before external users can see them.* On the **Page Templates** page, select and click the page name you want to administrate templates for. The **Available Templates** page for the selected template shows the templates currently in use for production and test purposes.

Attribute Search Page

The Attribute Search page displays when buyers start a product search.

Template	Description
Template 1	A thin border line outlines the Attribute Search area. Buttons appear at the bottom of the page only.
Template 2	This template is like the Default Template, except the Search, Reset and Back buttons appear both at the top and bottom of the page.
Default Template	The Attribute Search area appears without a border. Buttons appear at the bottom of the page only.

Model Comparison Page

The templates available for these pages are similar. Compare pages are used to do side-by-side product comparisons. Users can navigate to these pages from product list pages.

Template	Description
Template 1	This template displays product information in columns (one column for each product). The product name appears at top and product attributes appear below.
Template 2	This template is similar to Template 1, except product images appear above the columns.
Default Template	This template displays product information in rows (one row for each product). The product name displays at left. Product attributes display to the right.

Model List Page

The Model List page appears when a model search locates multiple models.

Template	Description
Template 1	This template displays products located from the search in generous sections, as wide as the page. Each section contains a product description, including images embedded in HTML, and the first catalog image. A thin border line outlines the sections.
Template 2	This template displays products located from the search in tiled sections, two sections per page width. Each section contains a product description, including images embedded in HTML, and the first catalog image. A thin border line outlines the sections.
Template 3	This template is similar to Template 2. The difference is catalog images do not display in the sections.
Template 4	This template is similar to Template 1. The difference is a border line does not outline the sections.
Template 5	This template is used to display a column-based results set. Product information appears in columns (one for each product). Images do not display with this template.
Template 6	This template is similar to Template 5, except a single catalog image appears at the top of each column.
Default Template	This template displays product information in rows (one row for each product). The product name displays at left. Product attributes display to the right.

Model Profile Page

The Model Profile page appears when buyers click a single model name from a list page, or when a single model results from a search.

Template	Description
Template 1	This template has three sections: top, middle and bottom. Catalog images and product attributes appear in the top section. HTML-formatted product descriptions appear in the middle section. The bottom section is reserved for associated files.
Template 2	This template has three sections: top, middle and bottom. The HTML-formatted product description appears top left, in a bordered area that also contains page buttons. The first catalog image displays top right. Associated files appear middle left. The second possible catalog image displays middle right. Product attributes span the bottom of the page.
Template 3	This template divides the page into three sections: top, middle and bottom. The HTML-formatted product description appears in a bordered area at the top of the page, along with page buttons. The middle section contains associated files. Product attributes span the bottom of the page.
Template 4	This template divides the page into three sections: top, middle and bottom. The HTML-formatted product description, including page buttons, spans the top of the page in a bordered area. The middle of the page shows product attributes at left and catalog images at right. The bottom right section displays associated files.
Template 5	This template divides the page into three sections: top left, top right and bottom. Product attributes appear at top left. Catalog images and associated files appear at top right. The HTML-formatted product description spans the bottom of the page in a bordered section.
Template 6	This template has three sections: top, middle and bottom. The HTML-formatted product description, including page buttons, spans the top of the page in a bordered section. Product attributes and catalog images appear in the middle region. The bottom section displays associated files.
Template 7	This template has just one section containing the HTML-formatted product description. The area is outlined with a thin border line.
Template 8 (product line)	This template has three sections: top, middle and bottom. The HTML-formatted product description, including page buttons, spans the top of the page. Product attributes and catalog images appear in the middle section. Associated files display at the bottom of the page.
Template 8 (model)	This template contains two sections: top and bottom. Product attributes appear in the top section, along with the first catalog image. Associated files appear in the bottom section, along with the second catalog image.
Template 9 (model)	This template is the same as Template 8 for product lines.
Default Template	This template divides the page into two sections: left and right. Product attributes appear on the left. Catalog images, the HTML-formatted product description and associated files appear from top to bottom on the right.

Part Associations Page

There is one default template available for the Part Display page and one XSL template that can be customized for the page. Users can navigate to the Part Display page after clicking on a part number from the Part Search Results page.

Template	Description
XSL Template	This template can be customized using XSL.
Default Template	This template displays part information at the top of the template and parent part information at the bottom of the template.

Part Display Page

There is one default template available for the Part Display page and one XSL template that can be customized for the page. Users can navigate to the Part Display page after clicking on a part number from the Part Search Results page.

Template	Description
XSL Template	This template can be customized using XSL.
Default Template	This template displays part information at the top of the template and parent part information at the bottom of the template.

Parts Search Page

The Parts Search area is the starting point for parts, serial number and keyword searches.

Template	Description
Template 1	This template displays both a part number search and a keyword search.
Template 2	This template displays a serial number search only.
Template 3	This template displays a part number search and serial number search, but no keyword search.
Template 4	This template displays a part number search only.
Default Template	This template displays all three search types: serial number, part number, and keyword.

Parts Search Results Page

There is one default template available for the Part Search Results page and one XSL template that can be customized for the page. Users reach the Part Search Results page after conducting a parts search.

Template	Description
XSL Template	This template can be customized using XSL.
Default Template	This template displays after conducting a parts search. Parts data displays in rows (one row for each part). The part name displays at left. Part description, lead time and price appear to the right.

Product Line Compare Page

The templates available for these pages are similar. Compare pages are used to do side-by-side product comparisons. Users can navigate to these pages from product list pages.

Template	Description
Template 1	This template displays product information in columns (one column for each product). The product name appears at top and product attributes appear below.
Template 2	This template is similar to Template 1, except product images appear above the columns.
Default Template	This template displays product information in rows (one row for each product). The product name displays at left. Product attributes display to the right.

Product Line List Page

The Product Line List page appears when a product line search locates multiple product lines.

Template	Description
Template 1	This template displays products located from the search in generous sections, as wide as the page. Each section contains a product description, including images embedded in HTML, and the first catalog image. A thin border line outlines the sections.
Template 2	This template displays products located from the search in tiled sections, two sections per page width. Each section contains a product description, including images embedded in HTML, and the first catalog image. A thin border line outlines the sections.
Template 3	This template is similar to Template 2. The difference is catalog images do not display in the sections.
Template 4	This template is similar to Template 1. The difference is a border line does not outline the sections.
Template 5	This template is used to display a column-based results set. Product information appears in columns (one for each product). Images do not display with this template.
Template 6	This template is similar to Template 5, except a single catalog image appears at the top of each column.
Default Template	This template displays product information in rows (one row for each product). The product name displays at left. Product attributes display to the right.

Product Line Profile Page

The Product Line Profile page appears when buyers click a single product name from a list page, or when a single product line results from a search.

Template	Description
Template 1	This template has three sections: top, middle and bottom. Catalog images and product attributes appear in the top section. HTML-formatted product descriptions appear in the middle section. The bottom section is reserved for associated files.
Template 2	This template has three sections: top, middle and bottom. The HTML-formatted product description appears top left, in a bordered area that also contains page buttons. The first catalog image displays top right. Associated files appear middle left. The second possible catalog image displays middle right. Product attributes span the bottom of the page.
Template 3	This template divides the page into three sections: top, middle and bottom. The HTML-formatted product description appears in a bordered area at the top of the page, along with page buttons. The middle section contains associated files. Product attributes span the bottom of the page.
Template 4	This template divides the page into three sections: top, middle and bottom. The HTML-formatted product description, including page buttons, spans the top of the page in a bordered area. The middle of the page shows product attributes at left and catalog images at right. The bottom right section displays associated files.
Template 5	This template divides the page into three sections: top left, top right and bottom. Product attributes appear at top left. Catalog images and associated files appear at top right. The HTML-formatted product description spans the bottom of the page in a bordered section.
Template 6	This template has three sections: top, middle and bottom. The HTML-formatted product description, including page buttons, spans the top of the page in a bordered section. Product attributes and catalog images appear in the middle region. The bottom section displays associated files.
Template 7	This template has just one section containing the HTML-formatted product description. The area is outlined with a thin border line.
Template 8 (product line)	This template has three sections: top, middle and bottom. The HTML-formatted product description, including page buttons, spans the top of the page. Product attributes and catalog images appear in the middle section. Associated files display at the bottom of the page.
Template 8 (model)	This template contains two sections: top and bottom. Product attributes appear in the top section, along with the first catalog image. Associated files appear in the bottom section, along with the second catalog image.
Template 9 (model)	This template is the same as Template 8 for product lines.
Default Template	This template divides the page into two sections: left and right. Product attributes appear on the left. Catalog images, the HTML-formatted product description and associated files appear from top to bottom on the right.

Recommended Items

The templates for the Recommended Items page provide users the ability to control the display of information on the model configuration/recommended items page. There is one default template available for the Recommended Items page and one custom template that can be customized for the page. Users reach the Recommended Items page from the model configuration/recommended items page.

Template	Description
Custom Template	This template displays all the columns from the parts and models in separate sections: part columns, model columns and rule columns. You can also re-label and reorder the columns. The ordering is done from left to right. For example, the column in the first position appears on the leftmost position on the model configuration/recommended items page, while the column in the last position appears on the rightmost position. If you delete a custom field from the parts section, it will get deleted from the templates section. In such a case, you will lose all permissions related data. If you add a custom field in parts, then this new custom field will be automatically display in the custom template section. This column will be visible to the end users, depending on the permissions set by you. When you choose the custom template, all columns are displayed to the end user, by default. You can then decide to hide certain columns. Columns can be always hidden or based on specific permissions. For more information on hiding columns based on permissions, read the section below on editing user permissions.
Default Template	This template displays after conducting a parts search. Parts data displays in rows (one row for each part). The part name displays at left. Part description, lead time and price appear to the right.

Serial Number Associations Page

There is one default template available for the Part Display page and one XSL template that can be customized for the page. Users can navigate to the Part Display page after clicking on a part number from the Part Search Results page.

Template	Description
XSL Template	This template can be customized using XSL.
Default Template	This template displays part information at the top of the template and parent part information at the bottom of the template.

Serial Number Search Results Page

There is one default template available for the Part Search Results page and one XSL template that can be customized for the page. Users reach the Part Search Results page after conducting a parts search.

Template	Description
XSL Template	This template can be customized using XSL.
Default Template	This template displays after conducting a parts search. Parts data displays in rows (one row for each part). The part name displays at left. Part description, lead time and price appear to the right.

NOTES



For parts pages **Part Display**, **Part Search Results**, and **Parts Search**, the system also supports XSL for customizing templates. For these pages, you can define custom page layouts using XSL Stylesheets. You can customize the XSL Stylesheet that defines the template before or after activating the template itself.



Most catalog pages support a standard set of system-defined templates which are static and not modifiable.

RELATED TOPICS

PUNCH-INS

OVERVIEW

Punch-ins are links that hook into specific pages in the system. They can be used to navigate directly from one page to another within your application. They can also be used to jump into a specific page from an external website. External punch-ins must contain the username and password information so that an authentication can be performed to ensure that a valid user is trying to access the page.

ADMINISTRATION

Creating a Custom Punch-In

1. Navigate from: Admin Home Page > Styles and Templates > Home Page.
2. Select the **product line/model** you need to create a punch-in for in the selector punch-in column.
3. Click the **New** button.
4. Enter the **name** and **variable name** for the punch-in.
5. To hide a punch-in on the home page, select the corresponding checkbox under the Hide column. Note: you can only hide default punch-ins.
6. You can set the following punch inputs in the parameters column:
    - To set attribute values on selector configuration or model configuration, use `variableName1=variableValue1&variableName2=variableValue2...`
    - To trigger conditions to pick up the right configuration flow in model configuration, use `variableName1+variableValue1+variableName2+variableValue2...`
    - To enable selector recommendations for model configuration, use `variableName1@variableValue1@variableName2@variableValue2...`
    - Change: `&_variable_name_punchin=true...`. This parameter controls whether to use the Label or variable name for parameters. It is defaulted to false.
7. Click on **Save** or **Save and Close** to save changes. Click on **Delete** to delete the selected custom punch-ins.
8. Deploy the homepage.



When the variable is of menu type, the menu values are passed as a tilde delimited string. When the variable is of array type, the array values are separated by "$,$".

USE CASES

Model Configuration Page Punch-In

`http://myCompany.bigmachines.com/commerce/new_equipment/products/model_configs.jsp?segment=Widgets&product_line-e=bigWidgets&model=widget1&bm_search_result=Direct%20Configured@Yes&_first_time=true`

Syntax:

- Parameters: `variableName1=variableValue1&variableName2=variableValue2`
- Flow Inputs: `variableName1+variableValue1+variableName2+variableValue2`
- Search Results: `variableName1@variableValue1@variableName2@variableValue2`

Model Configuration Punch-In (from external source)

`http://acme.bigmachines.com/commerce/model_detail.jsp?model_id=1224334&product_line_id=7082223&use-ername=tomsmith&psword=password`

Parts Display Page Punch-In

`http://myCompany.bigmachines.com/commerce/parts/display_part.jsp?part_number=00000049&company=4118175`

Selector Punch-In

`http://acme.bigmachines.com/commerce/new_equipment/products/selector_list.jsp?segment=Pumps&product_line=Godzilla%20Pumps&_bm_sel_punch_in_atts=A~10~color~red&from=product]`

Syntax:

- Parameters: `variableName1=variableValue1&variableName2=variableValue2`
- Flow Inputs: `variableName1+variableValue1+variableName2+variableValue2`
- Search Results: `variableName1@variableValue1@variableName2@variableValue2`

Array Attribute Punch-In

Use the separator $,$ to specify multiple values for array attributes in punch-ins. **Example:** when you want to set the values of attributes as follows: `size1 = 4` and `arr1[0] = v1`, `arr1[1]=v2`, `arr1[2]=`, `arr1[3]=v4` (here, `arr1[2]` is a blank value), the punch-in format will look like this ...`size1=4&arr1=v1$,$v2$,$,$v4&ldots;`

Example of a bm_search_result: `&ldots;bm_search_result= size1=4&arr1=v1$,$v2$,$,$v4`

NOTES



Attribute range calculations are validated on the Value Punch-In. **Example:** if the value was passed in the URL to the Configuration, it will be validated and an error message will appear. *This affects both*



Mobile and Desktop.

RELATED TOPICS

SITE BRANDING

OVERVIEW

Site branding allows companies to promote their brand and provide continuity to the user.

ADMINISTRATION

Applying precise branding

Branding your client's site is best handled when it is done precisely.

- **Match your company's specific branding; do not estimate.** ***Example:*** Your company wants their BigMachines site to look like their external website; their website has dark green text and light green tabs/buttons.

DO	DON'T

Obtain the precise branding colors for your client. **Do this properly by:**
• Use your company's styling information.
• **If unavailable to obtain styling information:**
  ○ Use Firebug to find the styles used.
  ○ Obtain a 1:1 screenshot of the copy site and utilize an eyedropper.
• Avoid using a color if the correct color cannot be located.
• Use the correct font weight and size (Note: some fonts cannot be viewed in BigMachines).]
• Ensure that the company color is appropriate for the situation (e.g. text).
	• In the CSS- pick the closest color that "looks right".
	• Type the name of the color into the CSS (e.g. color:green or background:magenta).

- **Be careful with branding and text.** The color red should never be used for main text color (even if your company utilizes red in their branding). *Red is generally reserved for errors and warnings.* Black or a very dark gray is recommended for your main text.
  - **Example:** Your company's main branding color is pink. They love the color pink and they even use it on the text all over their main site. *This is not considered BigMachines best practice.* Instead, use pink as a highlight color elsewhere in the BigMachines application and leave the main text black to contrast it.
- **Recreate your company's tabs and buttons precisely.** You may not want to just color in the BigMachines tabs and buttons with your company colors, if your company uses distinct tabs and buttons. In this case, replace the default BigMachines tabs and buttons with those used by your company.
- **Utilize repetition and contrast.** If two things are different in your application, make it obvious that they are different.

Limiting fonts and font sizes

When developing a site, it can be tempting to take advantage of font variety as a means to make sections appear less plain. This can include using more than one font family or more than one font size on a page or section. *Keep in mind the following best practices for font variation:*

- **Font Families:** Do not use more than two font families in one site. *If you use two different fonts, ensure that they are very different.*

- **Font Sizes:** Do not exceed three variations in font size for a single page. If you need to distinguish text from other text, consider changing font-weight or color to avoid a fourth font-size.
- Variety within text styling can help highlight particular content, add visual appeal, and make the content look more professional. Know when and how to vary your text.
    - **Font Family Example:** Your company's logo is in Arial, so you've used Arial font exclusively. After a review,you feel that something needs to change.

DO	DON'T
• Keep just the section headers in Arial (for logo continuity). • For the main interior text, pick a font that contrasts well w/ Arial (e.g. Garamond or Georgia).	• Utilize a font that is too similar to Arial (e.g. Helvetica). • Exceed two fonts.

    - **Font Size Example:**Commerce utilizes section headers, tab text, labels, values, line item label text, and line item value text, etc.

DO	DON'T
• Utilize the largest font size in the section headers. • Utilize middle-sized font for the tab text, labels, and values. • Utilize the smallest font size for line items and their column labels (differentiate them from one another by background color). • Differentiate the labels and values next to them by bolding the labels. • Utilize a maximum of three font sizes.	• Utilize a different font size for each different class of text. • Exceed three font size.

Limiting tabs to 1 row

Too many horizontal tabs can make a site look cluttered. **Example:** Configurators with 10 horizontal tabs may look fine on a high-res monitor with the browser maximized, but on a low-res device, the tabs may wrap or get cut off. A large number of tabs can be frustrating and cumbersome to the user. *Always keep the end-user in mind when limiting tabs.*

- **External users & sales reps:** Limit your tab use to six per row. *Sales reps or external users may not be as technical and could be more prone to missing information.* Create a better experience and limit your tabs by asking yourself key questions, such as:
    - Should any of these tabs be combined into one?
    - Are any of these tabs unnecessary?
    - Can some of these tabs be hidden?
- **Name and order your tabs appropriately:** Give your tabs a concise/informative title, put them in a logical order, and ensure their content is relevant and necessitates its own tab. This creates a good flow, which won't overwhelm the user in instances when there are many tabs.

NOTES



Images stored in the File Manager can be secured. For more information, see File Manager.

RELATED TOPICS

STYLING BEST PRACTICES

OVERVIEW

Navigate from: Admin Home Page > Style and Templates

ADMINISTRATION

Proper formatting

- Use Notepad++ or a CSS editor to write your custom CSS files. *This promotes proper syntax and improves formatting.*
- Use the tab key for indentation.
- Curly brackets should open at the end of a line with a selector, and close at the beginning of the line following the last property and value.
- Properly comment and organize your CSS when you make additions or changes.
- Beware of special characters. *Some characters are reserved in HTML, XML, or XHTML because the browsers will mix them with tags. So if you need to use a special character use the HTML Entity.*
    - **Reserved characters:** < > & "
    - HTML Entities can be numeric or named; they all start with "&" and end with ";".
    - Use an entity name instead of a number; it's easier to remember. ***Note:*** *some browsers may not support all entity names.*
    - Click here for a list of HTML character entities.
- Use the hex code value instead of typing a color name. *This ensures continuity for branding purposes.*
    - When removing a selector color: either comment out the property & value OR delete the entire line.
    - Never leave a value blank after the #.

GOOD	BAD

`.bundle-list {`
`  border-color: #000000;`
`  border-style: solid;`
`}`
	`.bundle-list {`
	`  border-color: black;`
	`  border-style: solid;`
	`}`
	OR
	`.bundle-list {`
	`  border-color: #;`
	`  border-style: solid;`
	`}`

Class and ID naming conventions

- Always use lower case.
- Always begin with a letter. *CSS rules allow you to begin with an _underscore, but this is not a best practice and should be avoided.*
- Do not use spaces. Separate words 'with-a-dash'
- No special characters.
- Use a short, descriptive name.
    - Create descriptive names that describe the function, not the appearance of the section.
    - **Example:** Name an element located on the right-hand side of the page but will hold a list of bundle names, "'bundle-list" instead of "right-hand-column."
- **ID:** use the variable name when possible.

Proper HTML tag usage

- **Headings:** <h1> through <h6>. *Avoid using heading tags simply to bold text or format something differently*.
    - **h1:** Intended to be the page/main subject title (think of this as a book title). *Search engines often use h1 tags when looking at your site; overusing h1 tags can flag your site as spam.*
    - **h2:** Intended for section titles (think of these as chapter titles).
    - **h3-h6:** Sub-section titles; these are intended to be used in descending order.
- **Paragraphs:**
    - **
 tag:** Forces a line break within the text flow of a paragraph to continue your text on a new line. *Avoid using <br/> to end a paragraph or to change the length of your text lines.*
    - Avoid using   to add more space between elements. *It can result in awkward-looking HTML, which is confusing to edit later, and unexpected spacing in different browsers.*
    - Use a class name with the formatting found in your style sheets to achieve your desired spacing amount and type (e.g. indent, padding, first line indent, etc.).
- **Text emphasis**
    - CSS is the best way to format your text.
        - Assign a class to the text element (e.g. span, cell, group, grid). *That class can be used to format font-weight, font-face, size, color, alignment, etc.*
        - However, if you need to assign the styling within the HTML, use the correct tags.

    - **Bold text** Use . *<b> is an outdated tag; it works for now, but may not be supported in the future.*
    - **Italic text:** Use . *<i>is an outdated tag.; it works for now, but may not be supported in the future.*
- **Class and ID tags**
    - Always use lower case.
    - Always begin with a letter.
    - Use a short, descriptive name. *Create descriptive names that describe the function, not the appearance of the section.*
    - Don't use spaces: separate words "with-a-dash."
    - IDs should be used only for unique elements.
    - Classes can be used multiple times.
    - Always close your tags. *Failing to close a tag results in unwanted formatting, or "worst case scenario," can break your site (by interfering with JavaScripts or making critical elements inaccessible, invisible, or incorrectly displayed).*
- Always close your tags. *Failing to close a tag results in unwanted formatting, or "worst case scenario," can break your site (by interfering with JavaScripts or making critical elements inaccessible, invisible, or incorrectly displayed).*
- Use lower case markup.
- Avoid inline styles. *Instead, utilize CSS to style a class.*
    - Never use inline JavaScript.
- Validate your HTML.

Styles can be applied to Classes, ID's, and HTML elements

- Classes can be reused as needed throughout your site.
    - **Config:** classes can be applied to Tabs and Grids (within "Properties"). *If a class name is not specified, BigMachines will automatically assign a Grids name, based on default functionality.*
    - **CSS:** a class name always begins with a "." (e.g. '.group-list').
    - Avoid assigning either too many or too few classes to your elements.
- ID's are a unique styling: generally they're used to designate a special element (e.g. Body or Navigation bar) that are only used once per page.
    - **CSS:** ID names always begins with a "#" (e.g. '#nav-bar').

- Elements are the standard HTML components. **Examples:**
    - h1 = Heading 1
    - p = paragraph
    - td = table cell
- When styling an element, avoid applying a style across an entire site if that style is only intended for use in one section or a specific item.

Specificity in CSS

Specificity is a concept in CSS that determines how a conflict is resolved (e.g. competing CSS rules). The concept of specificity states that when two/more declarations apply to the same element, set the same property, and have equal importance/origin, the declaration with the most specific selector takes precedence.

- Be as specific as possible with your selector when writing a new CSS rule.
- When there is a conflict, utilize Firebug to discover which rule is conflicting with yours.
- Avoid using the `'!important'` tag to resolve conflicts that can be more easily resolved utilizing the principles of specificity.

Specificity example

1. Create the following rule in your Alt Stylesheet to display Configurator tab content 900 pixels wide:

```
tab-content {
width: 900px;
}
```

2. Check Firebug; something is wrong. After the CSS has been applied, the tab content remains at 1000 pixels:

```
.body-inner .tab-content {
width: 1000px;
}
```

3. After utilizing Firebug, you find that the above rule has a higher specificity value (which is why it takes precedence over the rule you've added to the Alt Stylesheet).

DO	DON'T

Make your rule more specific; adding another rule in the hierarchy gives your rule a higher specificity value.
Example:
`.body-inner .configuration .tab-content {`
`width: 900px;`
`}`
	Insert an "`!important`" tag. *This overloads your CSS.*

Calculating specificity

1. If one declaration is from a style attribute, rather than a rule with a selector (which is considered an inline style), it has the highest specificity. *If none of the declarations are inline, proceed to the next step.*
2. Count the ID selectors. The declaration with the highest count has the highest specificity. *If two/more have the same number of ID selectors, or if they don't have any ID selectors, proceed to the next step.*
3. Count the class selectors (e.g. `.test`), attribute selectors (e.g. `[type="submit"]`), and pseudo-classes (e.g. `:hover`). The declaration with the highest total has the highest specificity. *If two/more have the same total, or have totals of zero, proceed to the next step.*
4. Count the element type selectors (e.g. `div`) and pseudo-elements (e.g. `:first-letter`). The declaration with the highest total has the highest specificity.



According to CSS rules, if two/more selectors have the same specificity, then the latter specified rule takes precedence.

Hiding elements using CSS

Before hiding elements using the CSS, ask yourself the following questions:

- *Is the element part of an Attribute or something that can be hidden using Hiding Rules?* If yes, utilize BigMachines rules instead. Hiding an element with CSS doesn't remove it from the HTML (therefore, can still be accessed).
- *Can the element be removed through the Layout Editor or by changing the flow?* If, yes, do it that way.

Hiding elements

1. Open Firefox Firebug.
2. Use Firebug's inspect tool to highlight and select the div/element you want hidden. *This will direct you to its HTML.*
3. Ensure that your selection isn't too broad or too specific. *What is highlighted with Firebug is what will be hidden.*
4. Note the element's class and/or ID within the HTML. *An ID is preferred, since it should only apply to this element; classes require more attention as a class may be used by other elements within the site.*
5. Create a rule in the CSS. *Make the selector as specific as possible.*
6. Place the line, `"display: none;"`, within the CSS rule. *Comment the rule appropriately so others know what is being hidden.*
7. Check your results and site to ensure the correct elements have been hidden.

Example

Your site has the drop-down menu, titled, "Price Book". Because this element does not serve a purpose on your site, you would like to hide it. Utilize Firebug to locate it's class and create the following CSS rule (with proper commenting):

```
/*Hiding Price Book Element*/
.pricebook-container{
display:none;
}
```



Always consult your team members before hiding anything with CSS.

Adding/editing pre-existing CSS

In the process of implementing, upgrading, or maintaining a client site, you will likely find yourself needing to make a change to an existing stylesheet.

- Before replacing an existing stylesheet with your edited version, download and save a copy of the current stylesheet.
- Search the stylesheet before adding your change to see if a declaration with the same exact selector already exists.
    - *If the exact selector already exists w/in the CSS:* Place your change into the declaration rather than create a duplicate, conflicting one. Place a dated comment next to your changed line.
    - *If the particular selector/selector group doesn't already exist in the CSS:* Add new code. For more information, see Stylesheet Manager.
- When adding brand new code (e.g. new selector/selector group) to the stylesheet, place it in the appropriate section of related content. *If no such section exists or it is an isolated situation, place your code at the bottom and comment out a new section for it.*
- Place a comment above the selector with a clear explanation of what the CSS rule is accomplishing (include your initials & date if there are multiple people working on the CSS). *You can group related code together with a comment above the first rule and the word `"BEGIN"` followed by a comment below the last rule with the word `"END"`.*
- Maintain proper spacing between rules and comments.
    - Each rule should be spaced by (at least) one empty line from other rules.
    - Comments should be placed directly above the rules they reference.
    - Grouped comments (`"BEGIN"` and `"END"`) should have an empty line between the first and last rules they group.

Example

You want to change the width of class `.tab-content` to 1000px.

1. Search the stylesheet and find the existing `.tab-content` code.
2. Add the width declaration along with a comment, like this:

```
.tab-content{
width:1000px; /*Changed from 900 - JD, 4/21/12*/
}
```

3. If a declaration for `.tab-content` rule doesn't exist: add your new width declaration in the existing rule instead of making a duplicate selector.



Don't add a new rule for the selector .tab-content at the bottom of the stylesheet without checking to see if it exists. Adding a duplicate rule with a different value could create issues regarding which declaration takes precedence.



Always take care when adding new code. Use detailed comments, space appropriately, avoid redundancy.



Uncommented and disorganized code will make it more difficult for your co-workers to find and understand your changes. *It may also be difficult for you to remember the changes you have made later.*

External resources & design principles

Design principles

The following are concepts from *The Non-designer's Design Book* by Robin Williams.

- Begin with the end in mind: understand the end goal of the user and the site, so that your design is focused on the best way to reach that end goal.
- Never sacrifice legibility or usability.
- Contrast: if two items are not exactly the same make them very different.
- Repetition: repeat some aspect of the design throughout each page and throughout the site.

- Alignment: every item should have a visual connection with everything else on the page.
- Proximity: group related items together.

External resource: websites

Websites
UX Design
Yahoo UX
W3 Schools
W3C Markup Validation Service
A List Apart
Smashing Magazine
Eric A. Meyer's blog
Lynda.com *(paid subscription)*

External resources: books

Books
The Non-designer's Design Book - Robin Williams
Designing with Web Standards - Jeffrey Zeldman
Developing with Web Standards - John Allsopp
A Project Guide to UX Design: For user experience designers in the field or in the making - Ross Unger & Carolyn Chandler

---

NOTES



Before changing/adding a stylesheet: download the existing/default stylesheet to your local computer.
This is helpful if you need to revert changes or refer to the standard options.

Firebug is an extremely useful tool for locating element that you wish to style.



Document your changes and reasoning for said changes.

RELATED TOPICS

BASE TEMPLATE MODEL HOME PAGE

OVERVIEW



ADMINISTRATION

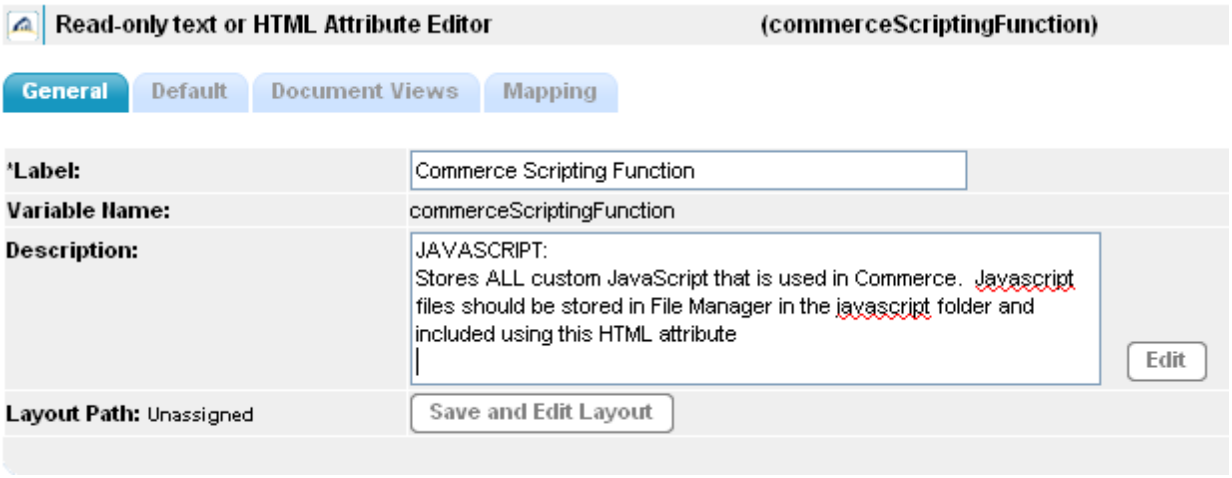
Setting up the Home Page

1. Navigate from: Admin > Home Page > Define XSL Templates.
2. Upload the attached files to the following respective areas:
    - XSL Files (Upload to Production and Internal Test): homepage.xsl
    - Alternate Stylesheet: AltStylesheetCSS
    - Alternate JavaScript File: homepage-altjs.js
3. Navigate from: Admin > Home Page > Deployment Center.
4. Deploy the Home Page.
5. Navigate from: Admin > File Manager.
6. Click on the JavaScript folder. *Create one if it doesn't exist.*
7. Load the file: gs_homepage.js to the folder.

Returning to Quote Setup from Home Page

Return to quote functionality allows users to return to the last open transaction from the Home Page. **How does it work?** The transaction ID, from the lasted visited transaction, is stored in a cookie. If the cookie is set, when the user navigates to the home page, the user sees a "Return to Quote" link. When they click the link, it opens up the transaction.

1. Under the desired Commerce Process, create an HTML Attribute named 'Commerce Scripting Function'. The Description should be as follows:



2. On the Default tab of the attribute, select the **Function** radio button.
3. Click **Define Function**.
4. Click **Next** since no attribute selection is necessary.
5. Enter the following code in the script definition area:

```
//Commerce Scripting HTML Attribute >> Default function
// script start and end tags
scriptStart = "<script type='text/javascript'>";
scriptEnd = "</script>";
// script include start and end tags
scriptIncludeStart = "<script type='text/javascript' src=";
scriptIncludeEnd = "></script>";
//Base path to file manager, allows relative call
fileManager = "$BASE_PATH$/";
returnStr = "";
// jQuery Cookie (jquery_cookie.js)
returnStr = returnStr + scriptIncludeStart + "'" + fileManager + "javascript/jquery_cookie.js'" + scriptIn-
cludeEnd;
// Return To Quote (qs_returnToQuote.js), requires jquery_cookie.js, tied to home page
returnStr = returnStr + scriptIncludeStart + "'" + fileManager + "javascript/qs_returnToQuote.js'" + scriptIn-
cludeEnd;
return returnStr;
```

6. Add the newly created attribute to the **Layout**. *Ensure that it is on a tab that is always first in the panel.*
7. Navigate to the **File Manager** and click on the JavaScript folder. *If it doesn't exist, create it.*

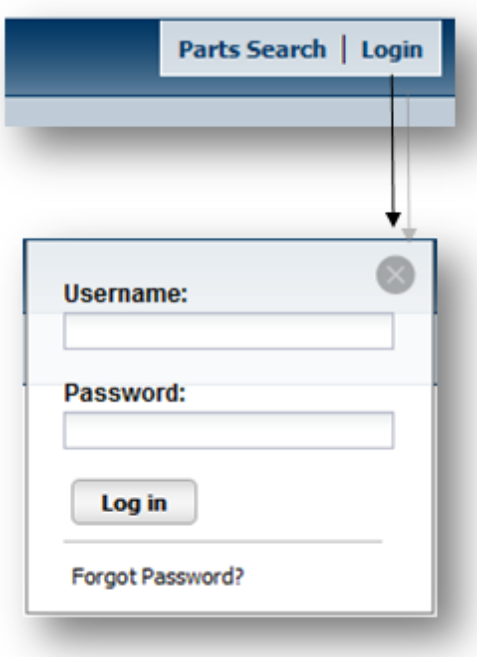
8. Load the file: return_to_quote.zip.

Login Screen



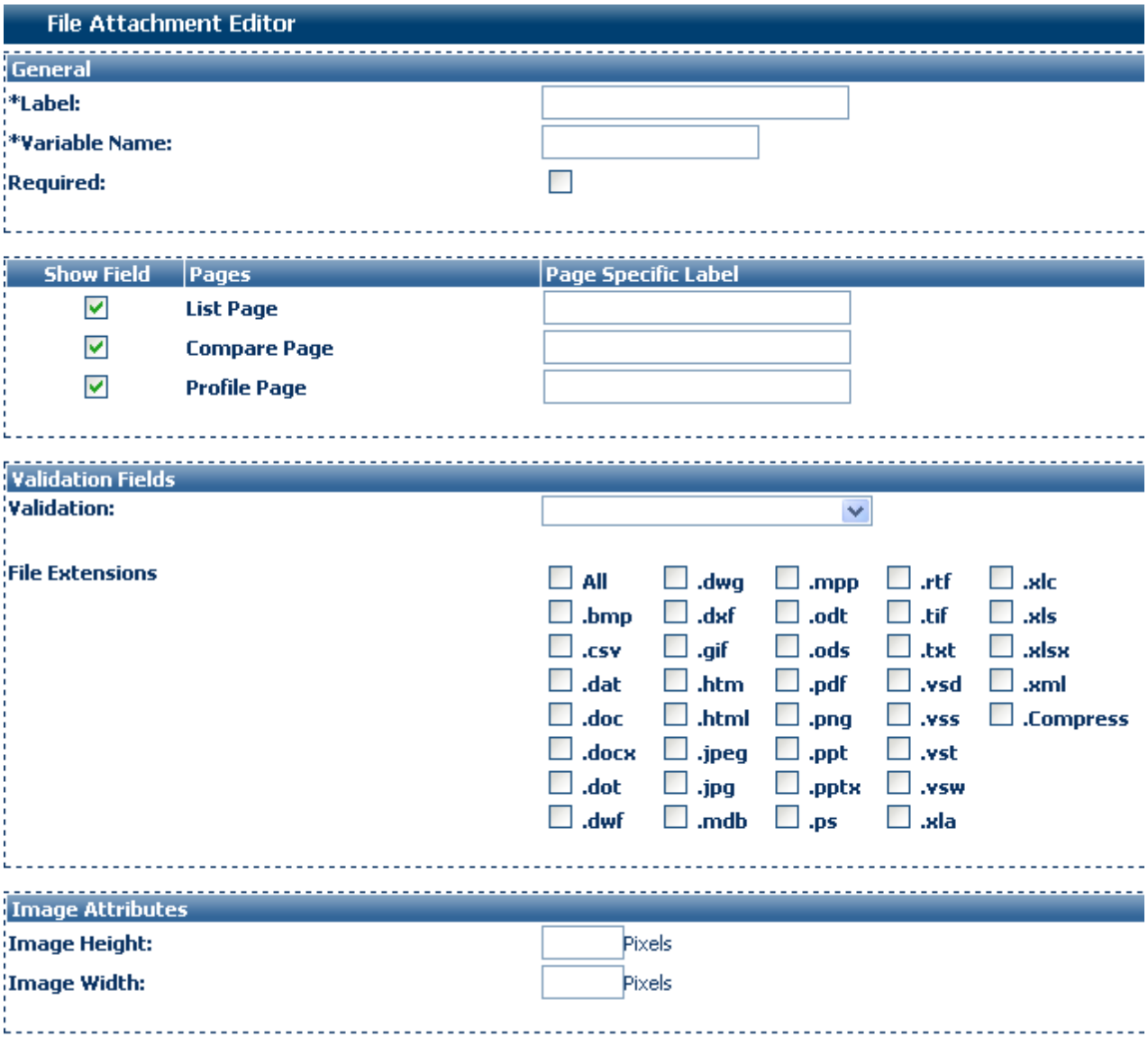
When Guest Access is enabled

- Login shows as a link
- Login Box is initially hidden
- User clicks link to expose the login box



Setting up the Alternate Home Page

1. Upload files to Admin Platform > Home Page > Define XSL Templates
2. Load the XSL, Alternate CSS and Alternate JavaScript files to their respective areas.
3. Deploy the **Home Page**.
4. Navigate to the **File Manager** and click on the JavaScript folder. *Create one if it doesn't exist.*
5. Load the qs_homepage.js file.
6. Create an image attribute for each product family model that is created. *Each file must have a valid extension.*



How displays are determined

- The Production XSL File determines what is displayed for most users.
- The Internal Test XSL File determines what is displayed for the Admin.
- The Alternate Stylesheet CSS file determines the styling specifically for the Home Page.



When new models are created, images should be loaded in addition to the description. To do so, navigate to the Models List page. For each page, click into the model name, upload the model image in the Image form field and click Update.



REQUIREMENT: JavaScript Library- load/implement the JS Framework first if you don't already have it.

NOTES



Alternate Stylesheet

RELATED TOPICS

BASE TEMPLATE MODEL HOME PAGE

OVERVIEW



ADMINISTRATION

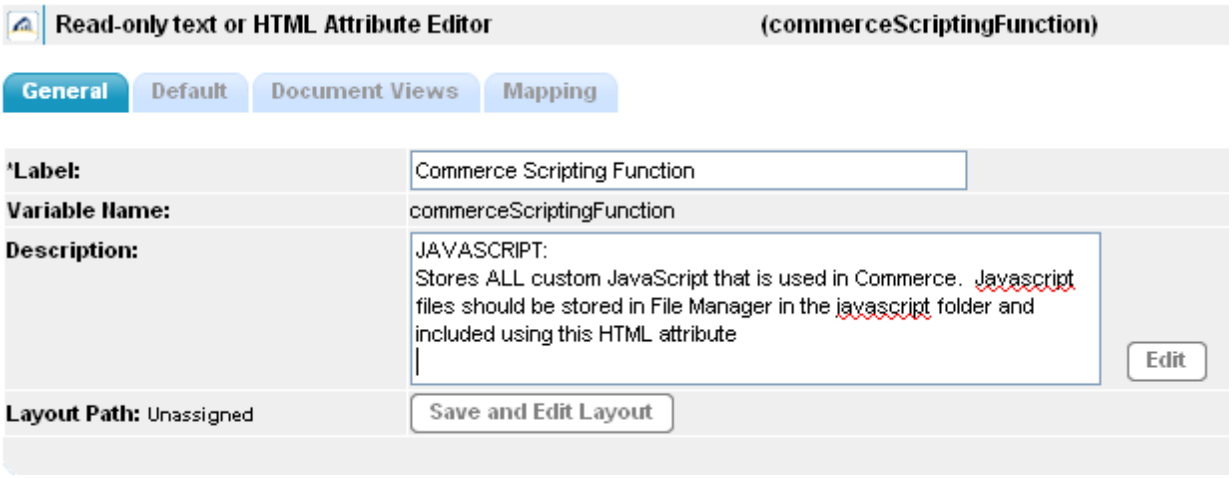
Setting up the Home Page

1. Navigate from: Admin > Home Page > Define XSL Templates.
2. Upload the attached files to the following respective areas:
    - XSL Files (Upload to Production and Internal Test): homepage.xsl
    - Alternate Stylesheet: AltStylesheetCSS
    - Alternate JavaScript File: homepage-altjs.js
3. Navigate from: Admin > Home Page > Deployment Center.
4. Deploy the Home Page.
5. Navigate from: Admin > File Manager.
6. Click on the JavaScript folder. *Create one if it doesn't exist.*
7. Load the file: gs_homepage.js to the folder.

Returning to Quote Setup from Home Page

Return to quote functionality allows users to return to the last open transaction from the Home Page. **How does it work?** The transaction ID, from the lasted visited transaction, is stored in a cookie. If the cookie is set, when the user navigates to the home page, the user sees a "Return to Quote" link. When they click the link, it opens up the transaction.

1. Under the desired Commerce Process, create an HTML Attribute named 'Commerce Scripting Function'. The Description should be as follows:



2. On the Default tab of the attribute, select the **Function** radio button.
3. Click **Define Function**.
4. Click **Next** since no attribute selection is necessary.
5. Enter the following code in the script definition area:

```
//Commerce Scripting HTML Attribute >> Default function
// script start and end tags
scriptStart = "<script type='text/javascript'>";
scriptEnd = "</script>";
// script include start and end tags
scriptIncludeStart = "<script type='text/javascript' src=";
scriptIncludeEnd = "></script>";
//Base path to file manager, allows relative call
fileManager = "$BASE_PATH$/";
returnStr = "";
// jQuery Cookie (jquery_cookie.js)
returnStr = returnStr + scriptIncludeStart + "'" + fileManager + "javascript/jquery_cookie.js'" + scriptIn-
cludeEnd;
// Return To Quote (qs_returnToQuote.js), requires jquery_cookie.js, tied to home page
returnStr = returnStr + scriptIncludeStart + "'" + fileManager + "javascript/qs_returnToQuote.js'" + scriptIn-
cludeEnd;
return returnStr;
```

6. Add the newly created attribute to the **Layout**. *Ensure that it is on a tab that is always first in the panel.*
7. Navigate to the **File Manager** and click on the JavaScript folder. *If it doesn't exist, create it.*

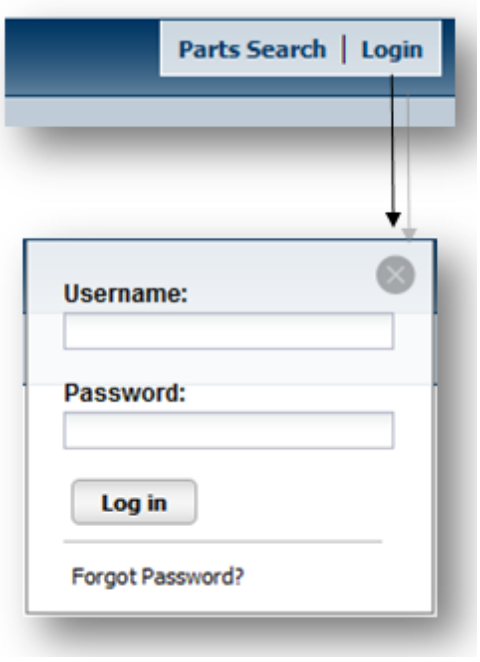
8. Load the file: return_to_quote.zip.

Login Screen



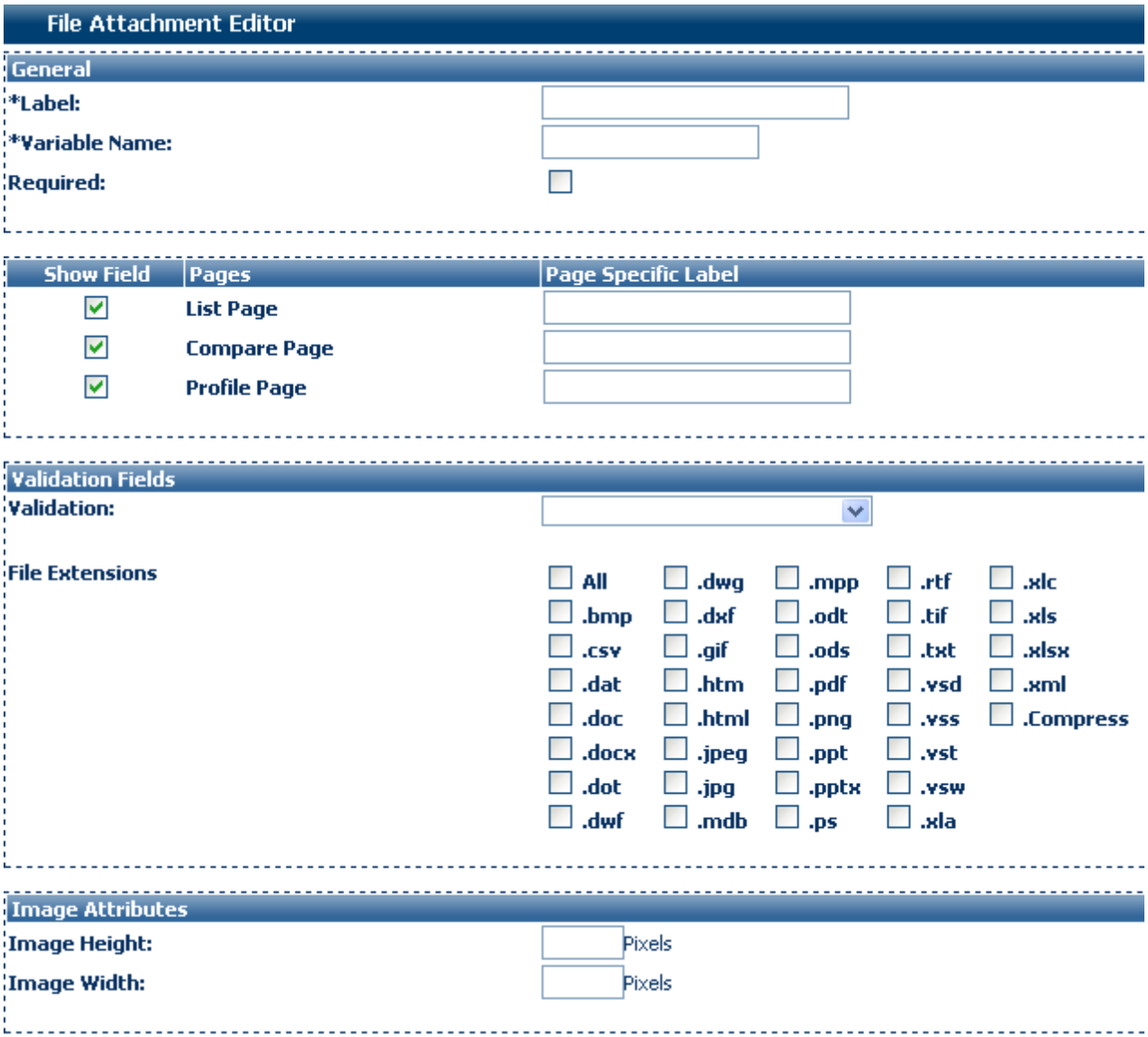
When Guest Access is enabled

- Login shows as a link
- Login Box is initially hidden
- User clicks link to expose the login box



Setting up the Alternate Home Page

1. Upload files to Admin Platform > Home Page > Define XSL Templates
2. Load the XSL, Alternate CSS and Alternate JavaScript files to their respective areas.
3. Deploy the **Home Page**.
4. Navigate to the **File Manager** and click on the JavaScript folder. *Create one if it doesn't exist.*
5. Load the qs_homepage.js file.
6. Create an image attribute for each product family model that is created. *Each file must have a valid extension.*



How displays are determined

- The Production XSL File determines what is displayed for most users.
- The Internal Test XSL File determines what is displayed for the Admin.
- The Alternate Stylesheet CSS file determines the styling specifically for the Home Page.



When new models are created, images should be loaded in addition to the description. To do so, navigate to the Models List page. For each page, click into the model name, upload the model image in the Image form field and click Update.



REQUIREMENT: JavaScript Library- load/implement the JS Framework first if you don't already have it.

NOTES



Alternate Stylesheet

RELATED TOPICS

BASE TEMPLATE MODEL HOME PAGE

OVERVIEW



ADMINISTRATION

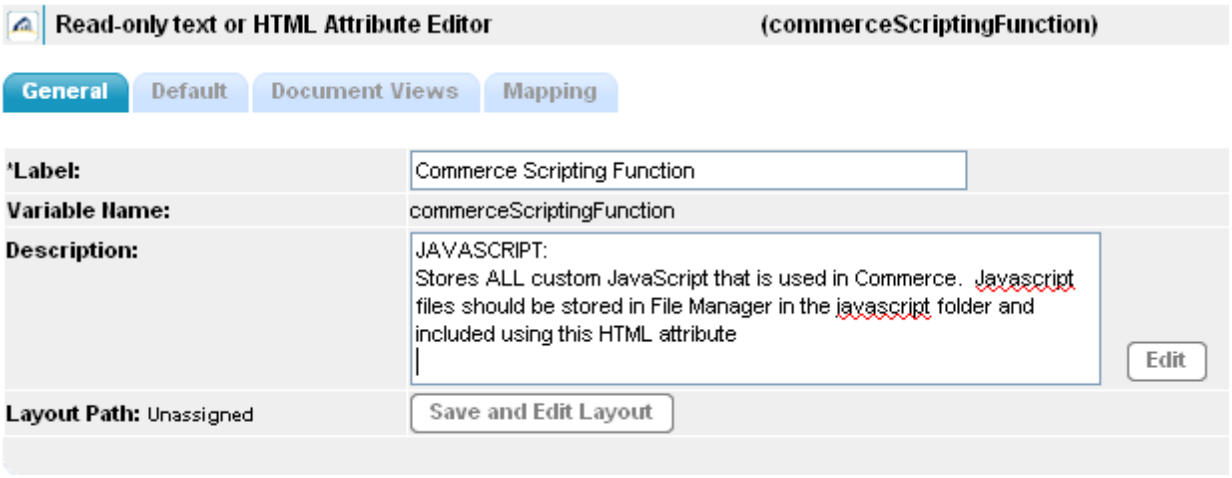
Setting up the Home Page

1. Navigate from: Admin > Home Page > Define XSL Templates.
2. Upload the attached files to the following respective areas:
    - XSL Files (Upload to Production and Internal Test): homepage.xsl
    - Alternate Stylesheet: AltStylesheetCSS
    - Alternate JavaScript File: homepage-altjs.js
3. Navigate from: Admin > Home Page > Deployment Center.
4. Deploy the Home Page.
5. Navigate from: Admin > File Manager.
6. Click on the JavaScript folder. *Create one if it doesn't exist.*
7. Load the file: gs_homepage.js to the folder.

Returning to Quote Setup from Home Page

Return to quote functionality allows users to return to the last open transaction from the Home Page. **How does it work?** The transaction ID, from the lasted visited transaction, is stored in a cookie. If the cookie is set, when the user navigates to the home page, the user sees a "Return to Quote" link. When they click the link, it opens up the transaction.

1. Under the desired Commerce Process, create an HTML Attribute named 'Commerce Scripting Function'. The Description should be as follows:



2. On the Default tab of the attribute, select the **Function** radio button.
3. Click **Define Function**.
4. Click **Next** since no attribute selection is necessary.
5. Enter the following code in the script definition area:

```
//Commerce Scripting HTML Attribute >> Default function
// script start and end tags
scriptStart = "<script type='text/javascript'>";
scriptEnd = "</script>";
// script include start and end tags
scriptIncludeStart = "<script type='text/javascript' src=";
scriptIncludeEnd = "></script>";
//Base path to file manager, allows relative call
fileManager = "$BASE_PATH$/";
returnStr = "";
// jQuery Cookie (jquery_cookie.js)
returnStr = returnStr + scriptIncludeStart + "'" + fileManager + "javascript/jquery_cookie.js'" + scriptIn-
cludeEnd;
// Return To Quote (qs_returnToQuote.js), requires jquery_cookie.js, tied to home page
returnStr = returnStr + scriptIncludeStart + "'" + fileManager + "javascript/qs_returnToQuote.js'" + scriptIn-
cludeEnd;
return returnStr;
```

6. Add the newly created attribute to the **Layout**. *Ensure that it is on a tab that is always first in the panel.*
7. Navigate to the **File Manager** and click on the JavaScript folder. *If it doesn't exist, create it.*

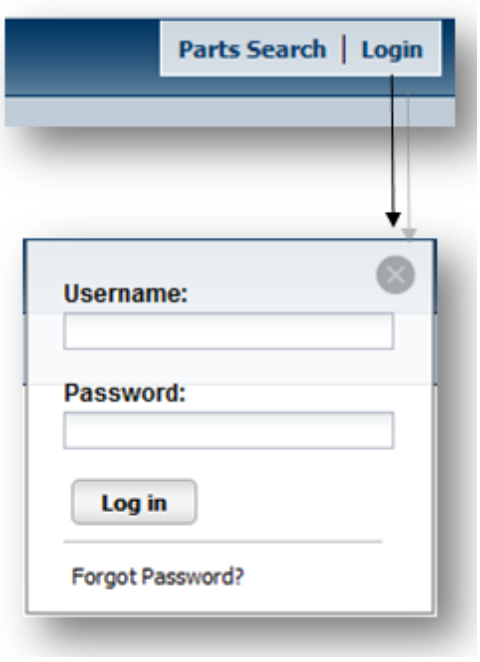
8. Load the file: return_to_quote.zip.

Login Screen



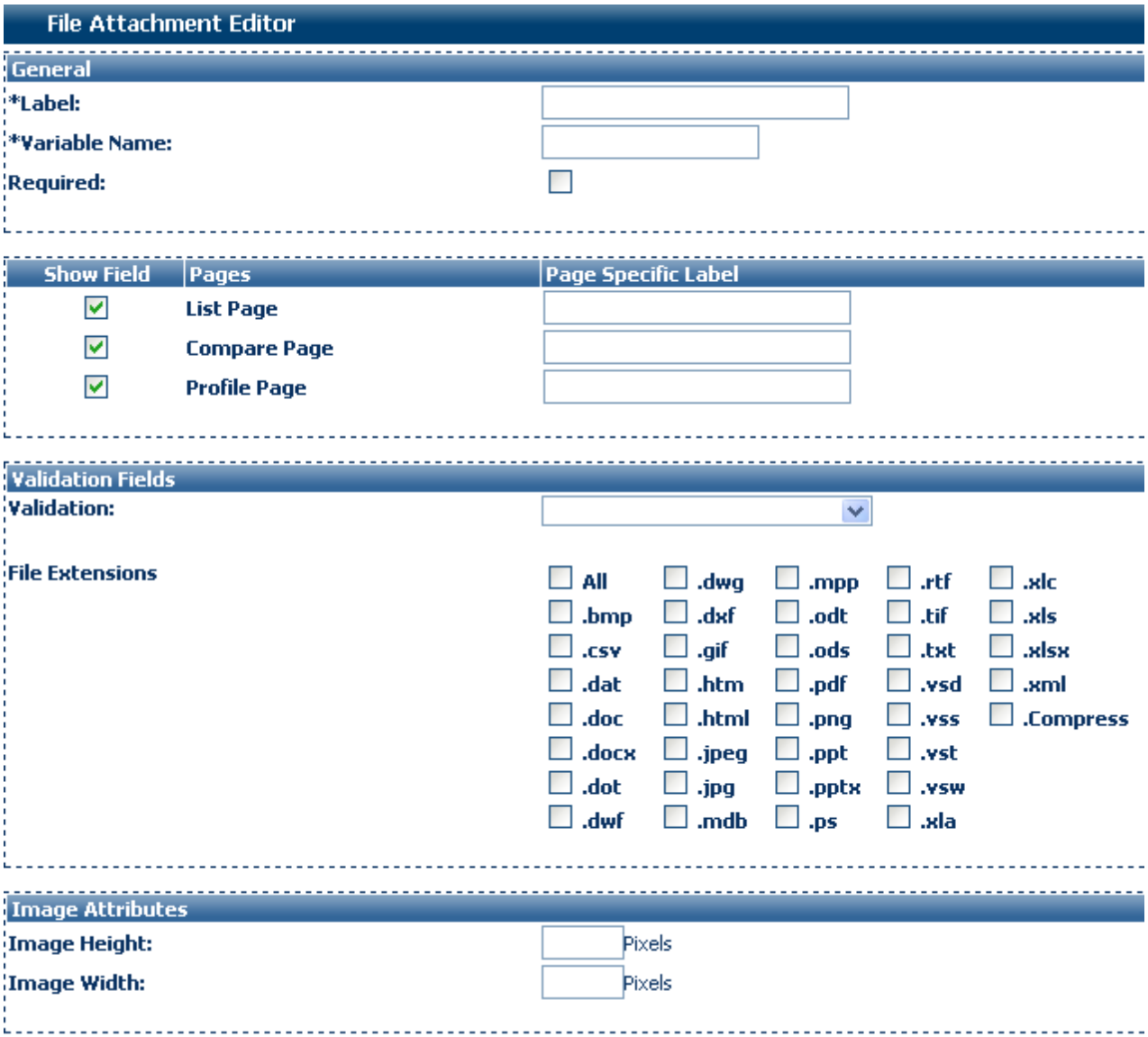
When Guest Access is enabled

- Login shows as a link
- Login Box is initially hidden
- User clicks link to expose the login box



Setting up the Alternate Home Page

1. Upload files to Admin Platform > Home Page > Define XSL Templates
2. Load the XSL, Alternate CSS and Alternate JavaScript files to their respective areas.
3. Deploy the **Home Page**.
4. Navigate to the **File Manager** and click on the JavaScript folder. *Create one if it doesn't exist.*
5. Load the qs_homepage.js file.
6. Create an image attribute for each product family model that is created. *Each file must have a valid extension.*



How displays are determined

- The Production XSL File determines what is displayed for most users.
- The Internal Test XSL File determines what is displayed for the Admin.
- The Alternate Stylesheet CSS file determines the styling specifically for the Home Page.



When new models are created, images should be loaded in addition to the description. To do so, navigate to the Models List page. For each page, click into the model name, upload the model image in the Image form field and click Update.



REQUIREMENT: JavaScript Library- load/implement the JS Framework first if you don't already have it.

NOTES



Alternate Stylesheet

RELATED TOPICS

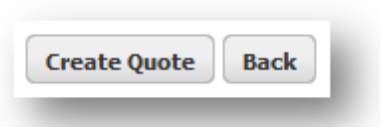
BASE TEMPLATE CSS

OVERVIEW

The following Commerce styling modifications require the utilization of the alternate stylesheet (attached below).

ADMINISTRATION

Buttons



Setting up Buttons

1. Navigate from: Admin > Stylesheet.
2. Download the **Alternate Stylesheet**. *If one doesn't exist, create a new file and name it* `alt.css`.

> If one already exists, there is a possibility that some of the data contained within it may cause this functionality to behave unexpectedly.

3. Add the following code to the CSS file and upload the file to Alternate CSS File. *See attached file below.*

```
/* Buttons */
/* ------------------------------------------------------ */
/* Hide Button Images
.button-middle a img {
display: none;
}
.button-middle a {
padding: 0!important;
}
Hide Button Images end */
/* Format Button Images */
.button-middle a img {
margin: -8px 0 0 2px; /* The top margin is minus half height of the image (16px). */
position: absolute;
top: 50% !important;
}
.button-middle a {
```

```
padding: 0 0 0 0px;
background-position: 0 center;
background-repeat: no-repeat;
}
.add-row-btn .button-middle a {
background-image: url("image/Icons/icon-plus.png");
padding-left: 20px;
}
/* Format Button Images end*/
.button-left, .button-right {
display: none!important; /* hide button sides used in original table buttons */
}
td.button-middle {
border: solid #999999 1px; /* Button Color */
background: #CCCCCC url("image/images/sw_navbargrad.png") repeat-x 0 0; /* Button Style*/
-webkit-background-clip: padding-box ; /* Fixes border bleed in webkit */
background-clip: padding-box;
-moz-background-clip: padding;
display: block;
position: relative;
-moz-border-radius: 2px;
-webkit-border-radius: 2px;
border-radius: 2px;
height: 26px;
margin: 0px;
}
.button-middle a {
outline: none;
-moz-outline-style: none;
margin: 0px 6px;
text-align: center;
font-family: Arial,sans-serif!important; /* Primary Font */
font-size: 12px!important;
text-transform: uppercase;
font-weight: normal!important;
color:#000000!important;
line-height: 26px;
display: block;
}
td.button-middle:active a{
```

```
margin: 0px 6px!important;
}
.button-middle div {
margin: 0px!important; /*overrides .js inline click movement */
padding: 0!important;
}
.button-text, .button-text:link, .button-text:visited {
position: relative;
top: 0px;
}
#sticky-actions td.button-middle {
border: solid #999999 1px; /* Button Color */
background: #999999 url("image/images/sw_navbargrad.png") repeat-x 0 0; /* Button Style*/
height: 30px;
margin: 0px;
}
#sticky-actions .button-middle a {
margin: 0px 8px;
font-family: Arial,sans-serif!important; /* Primary Font */
font-size: 12px!important;
text-transform: uppercase;
font-weight: normal!important;
color:#000000!important;
line-height: 30px;
}
#sticky-actions td.button-middle:hover,
td.button-middle:hover {
background-color: #C3DDE3; /* Accent Color */
border-color: #4B626C; /* Tertiary Color */
}
#sticky-actions .button-middle:hover a,
.button-middle:hover a{
color: #333333!important;
}
#sticky-actions td.button-middle:active a{
margin: 0px 8px!important;
}
#sticky-actions td.button-middle:active ,
td.button-middle:active {
background-color: #4B626C;
```

```
}
/* Call-to-action Button color */
.invocation-button td.button-middle{
background-color: #86B5A2!important;
}
.invocation-button td.button-middle:hover{
background-color: #B4F0D8!important;
}
table[onclick="javascript:setDocFormIds(5003735, 1, 5004393);submitFormActionEvent('/commerce/buyside/document.jsp', false);bmCancelBubble(event)"] td.button-middle{
background-color: #86B5A2!important;
}
table[onclick="javascript:setDocFormIds(5003735, 1, 5004393);submitFormActionEvent('/commerce/buyside/document.jsp', false);bmCancelBubble(event)"] td.button-middle:hover{
background-color: #B4F0D8!important;
}
/* Commerce Buttons with Icons
a#return_to_opportunity
a#view_revision_history
a#save
a#add_line_items
a#copy_line_items
a#delete_selected
a#submit
*/
```

4. Click **Accept**.
5. Navigate to **File Manager** and select the **Images** folder. *If this doesn't exist, create it.*
6. Load the file `button-back.png` onto the folder. *See attached file.*



The `border-radius line` tags determine the roundness of a button's corners. *Deleting these lines will create perfect rectangles. Increasing the radius of these lines will make the buttons rounder.*

Setting up Buttons with Icons

This add icons to the: **Refresh**, **Restart**, **Add To Quote**, and **Add To Cart** buttons.

1. Navigate from: Admin > Stylesheet.
2. Download the **Alternate Stylesheet**. *If one doesn't exist, create a new one.*



If one already exists, there is a possibility that some of the data contained within it may cause this functionality to behave unexpectedly.

3. Add the following code to the CSS button file and upload the file to Alternate CSS File.

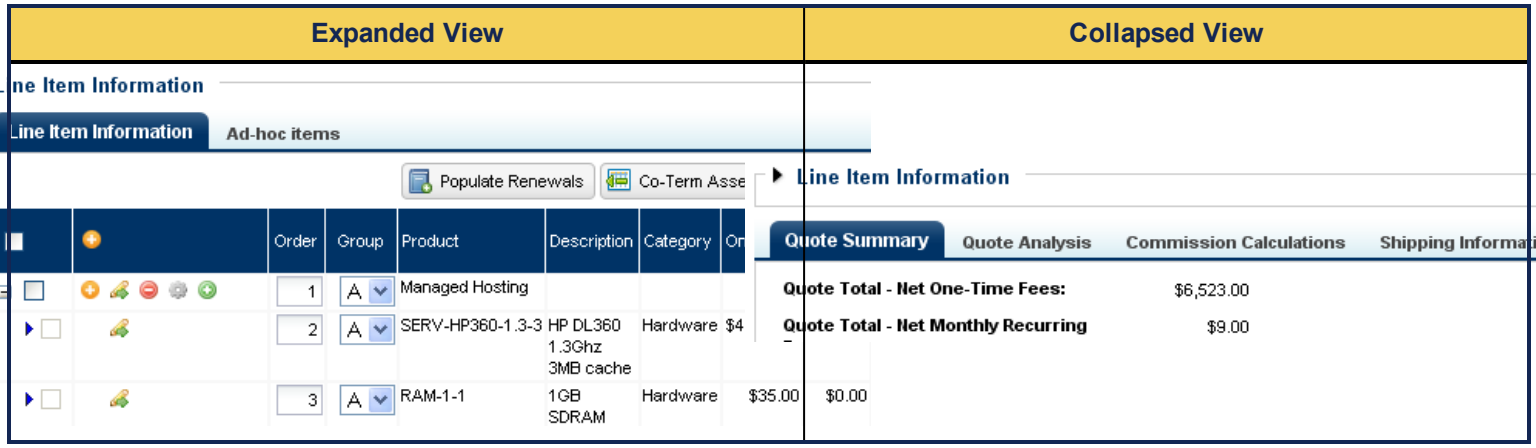
```
/* Commerce Buttons with Icons
a#return_to_opportunity
a#view_revision_history
a#save
a#add_line_items
a#copy_line_items
a#delete_selected
a#submit
*/
```

4. Upload the file to Alternate CSS file.
5. Click **Accept**.
6. Navigate to the **File Manager**and select the **Images** folder.
7. Upload the file `configuration_icons.zip`.

Modifying Button properties

1. To modify the button color, search for the following: *(for all of the below, only the background-color attribute needs to be modified)*
    - **Color:** td.button-middle
    - **Hover-Color:** td.button-middle:hover
    - **Active Color:** td.button-middle:active
2. To modify the height of the button:
    - Modify the line-height attribute: .button-middle a

Panels



Setting up Panels

1. Navigate from: Admin > Stylesheet.
2. Download the **Alternate Stylesheet**. *If one doesn't exist, create a new one.*

> If one already exists, there is a possibility that some of the data contained within it may cause this functionality to behave unexpectedly.

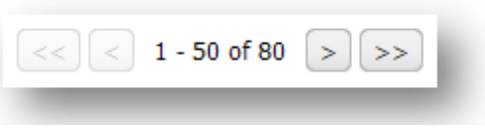
3. Add the following code to the CSS file.

```
/* Panels
------------------------------------------------------ */
.panel .panel-body { padding: 10px; border: none; }
.tab-content { border: solid #ededed; border-width: 0 1px 1px 1px; padding: 10px; }
.panel.collapsible { padding: 15px 5px; border: solid 1px #aaa; margin: 30px 15px 0 5px; }
div.panel .panel-tl,
div.panel .panel-tc,
div.panel .panel-tr { background: #fff; width: auto !important; }
div.panel .panel-tl {
border: none;
margin: -28px 0 0 5px;
padding: 0 5px 0 30px;
float: left;
background-image: url(image/images/arrow-expanded.png);
background-repeat: no-repeat;
background-position: 8px 4px ;
}
```

```
div.panel-bl { display: none; }
div.panel-collapsed .panel-tl { background-image: url(image/images/arrow-collapsed.png); }
div.panel .panel-heading { color: #333; font-size: 14px; }
.toggle-panel { display: none; }
```

4. Upload the file under Alternate CSS file.
5. Click Accept.
6. Navigate to the File Manager and upload the file panels.zip to the images folder.

Paging



Setting up Paging

1. Navigate from: Admin > Stylesheet.
2. Download the **Alternate Stylesheet**. *If one doesn't exist, create a new one.*

> If one already exists, there is a possibility that some of the data contained within it may cause this functionality to behave unexpectedly.

3. Add the following code to the CSS file.

```
/* Pagination */
/* ------------------------------------------------------ */
.column .paging-tools{
text-align: center;
font-size: 12px;
}
.paging-tools {
margin-top: 15px;
}
div.paging-tools span{
visibility: hidden;
padding: 0;
}
```

```
.paging-tools .first-page,
.paging-tools .prev-page,
.paging-tools .paging-status,
.paging-tools .next-page,
.paging-tools .last-page {
visibility: visible;
}
.paging-status {
margin: 0 10px 0 6px;
}
.paging-tools a.paging-tool-disabled {
color: #999999!important;
}
.paging-tools span a {
padding: 3px 6px;
text-decoration: none;
font-family: Arial,sans-serif!important; /* Primary Font */
color:#000000!important;
-moz-border-radius: 2px;
-webkit-border-radius: 2px;
border-radius: 2px;
-webkit-background-clip: padding-box; /* Fixes border bleed in webkit */
background-clip: padding-box !important;
-moz-background-clip: padding !important;
border: solid #999999 1px; /* Button Color */
background: #999999 url("image/images/sw_navbargrad.png") repeat-x 0 0; /* Button Style*/
}
.paging-tools span a:hover {
background-color: #C3DDE3; /* Accent Color */
border-color: #4B626C; /* Tertiary Color */
color: #333333!important;
}
.paging-tools span a:active {
background-color: #4B626C; /* Tertiary Color */
background-position: 0 -310px;
}
```

4. Upload the file to Alternate CSS file.
5. Click Accept.
6. Navigate to the File Manager and upload the file button_back.png to the images folder.

Setting up Line Item Grid

1. Navigate from: Admin > Stylesheet.
2. Download the **Alternate Stylesheet**. *If one doesn't exist, create a new one.*

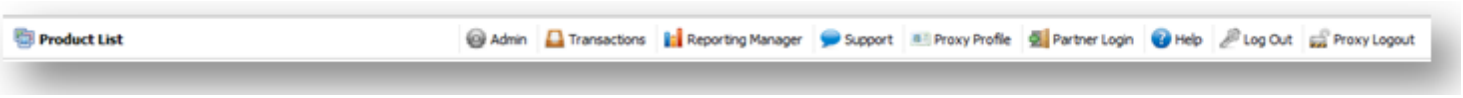
> If one already exists, there is a possibility that some of the data contained within it may cause this functionality to behave unexpectedly.

3. Add the following code to the CSS file.

```
/* Line Item Grid
------------------------------------------------------- */
#line-item-grid tr.parent-line-item > td {
padding: 3px 5px;
line-height: 18px !important;
height: 18px !important;
border-right: solid 1px #eee;
}
#line-item-grid .parent-line-item:hover td { background-color: #f1f3f6; }
#line-item-grid { border: #eee solid; border-width: 0 0 0 1px; }
.line-item-grid-header th { font-size: 12px; padding: 4px 2px;}
.line-item-grid-footer {}
.line-item-grid-footer th{ border-width: 4px 0; padding: 3px 5px; font-size: 14px; }
td.checker{ text-align: center; }
```

4. Upload the file to the Alternate CSS.
5. Click **Accept**.

Navigation Menu



Customizing the Navigation Menu

1. Navigate from: Admin > Navigation Menus
2. Click **List Links** to select the Navigation Menu.

3. Click **Customize Menu**.
4. For all Company Type and User Type Combinations:
    - Put Home link in the far left column and make the width 25%. *If Home link doesn't exist, make the column 0%*
    - Put all other links in the far right column and make the width 75%. *If Home link doesn't exist, make the column 100%*
    - There should be no links in the middle column. *Make sure the width is 0%.*
5. Click **Apply** after each Company/User Type combination.
6. Click **Update** when this task is completed.
7. Click **Back** and **Edit Stylesheet**.
8. Download the **Alternate Stylesheet**. *If one doesn't exist, create a new one.*



If one already exists, there is a possibility that some of the data contained within it may cause this functionality to behave unexpectedly.

9. Add the following code to the CSS file.

```
/* ---------------Navigation Menu Stylesheet Alt------------------- */
#login-form {
border-radius: 5px;
-moz-border-radius: 5px;
box-shadow: none;
-moz-box-shadow: none;
/* box-shadow: 1px 1px 4px #888;
-moz-box-shadow: 1px 1px 4px #888;*/
font-family: Arial,sans-serif;
}
.header-char-delim,
.admin-char-delim,
.commerce-char-delim{
display: none;
}
#breadcrumbs{
display:none;
}
```

```
.header-bordercolor .nav-links a.commerce-sidebar-item,
.header-bordercolor .nav-links a.commerce-sidebar-current,
.header-bordercolor .nav-links a.admin-sidebar-item,
.header-bordercolor .nav-links a.admin-sidebar-current,
.header-bordercolor .nav-links a.header-item,
.header-bordercolor .nav-links a.header-current {
text-decoration: none;
outline: none;
}
.nav-left {
padding-left: 0px;
}
.nav-right {
padding-right: 0;
}
.header-bordercolor {
border: 1px solid #AAAAAA;
}
.admin-bordercolor,
.commerce-bordercolor{
border: 1px solid #999999;
height: 28px;
}
.header-bordercolor .nav-links{
padding: 0;
height: 28px;
}
.header-bordercolor tbody{
background: url("../../image/images/sw_navbargrad.png") repeat-x 0 0 #AAAAAA!important;
height: 28px;
}
.admin-bordercolor tbody,
.commerce-bordercolor tbody{
background: #CCCCCC!important;
height: 28px;
}
.admin-bordercolor nobr,
.commerce-bordercolor nobr {
border-left: 1px solid #999999;
```

```
border-right: 1px solid #999999;
margin: 0 -2px !important;
padding: 0 2px 0 5px;
line-height: 28px;
display: inline-block;
}
.header-bordercolor nobr{
border-left: 1px solid #BBBBBB;
border-right: 1px solid #BBBBBB;
margin: 0 -2px !important;
line-height: 28px;
display: inline-block;
padding: 0 2px 0 5px;
}
.admin-bordercolor td,
.commerce-bordercolor td,
.header-bordercolor td,
.nav-links,
.admin-bordercolor table,
.commerce-bordercolor table,
.header-bordercolor table {
white-space: nowrap;
padding:0px!important;
}
.commerce-bordercolor nobr:hover,
.admin-bordercolor nobr:hover,
.header-bordercolor nobr:hover {
color: #000000;
background: #EEEEEE;
}
.header-item,
.header-item:link,
.header-item:visited
.admin-sidebar-item,
.admin-sidebar-item:link,
.admin-sidebar-item:visited {
margin: 0 3px 0 0;
}
.sidebar-folder a img {
```

```
margin: 0;
}
/* -------- Set all Nav fonts to Arial ---------*/
.admin-sidebar-item,
.admin-sidebar-item:link,
.admin-sidebar-item:visited,
.admin-sidebar-current,
.admin-nav-tab,
.admin-nav-tab-on ,
.admin-tabs .tab-text,
.admin-char-delim ,
.commerce-sidebar-item,
.commerce-sidebar-item:link,
.commerce-sidebar-item:visited,
.commerce-sidebar-current,
.commerce-nav-tab, .commerce-nav-tab-on,
.commerce-tabs .tab-text,
.commerce-char-delim ,
.header-item,
.header-item:link,
.header-item:visited,
.header-current,
.header-tab,
.header-tab-on,
.header-tabs .tab-text,
.header-char-delim,
.footer-item,
.footer-item:link,
.footer-item:visited,
.footer-current,
.footer-tab, .footer-tab-on,
.footer-tabs .tab-text,
.footer-char-delim,
#main-nav,
#main-nav a,
#footer-nav,
#footer-nav a,
.admin-nav,
.admin-nav a,
```

```
.user-nav,
.user-nav a,
#main-nav .tab-text,
#footer-nav .tab-text,
.admin-nav .tab-text,
.user-nav .tab-text,
#login-form {
font-family: Arial,sans-serif;
}
```

10. Upload file to Alternate CSS File. *Don't forget to click **Update** so that file is uploaded.*
11. Click **Back**.
12. Click **List Links for Subheader**.
13. Click **Customize Menus**.

> Verify that for every user type Home is in the right column and all other links are in the left column. There should be no links in the middle column. Ensure that for every Navigation Menu type, the links are in the correct column.

14. Click **Back** (twice).
15. Click **Deploy**.

Tabs



Setting up Tabs

1. Navigate from: Admin > Stylesheet.
2. Download the **Alternate Stylesheet**. *If one doesn't exist, create a new one.*



If one already exists, there is a possibility that some of the data contained within it may cause this functionality to behave unexpectedly.

3. Add the following code to the CSS file.

```
/* Tabs
----------------------------------------------------- */
/* Custom Tabs */
.page-tabs.tab-strip {
background: #fff url(image/images/light-tab-background.png) left -1px repeat-x !important;
padding: 0 0 0 10px;
border-bottom-width: 2px;
border-bottom: 1px #003366 solid;
}
.tab-inner { background: none !important; }
.tab-right { background: none !important; }
.tab-strip .tab-text {
padding: 0;
color: #444;
padding-top: 3px;
}
.tab-strip a.tab-left:hover .tab-text { color: #000; }
.tab-strip .tab-active .tab-text { padding: 0; font-size: 12px; }
.tab-strip .tab-active a.tab-left,
.tab-strip .tab-active a.tab-left:hover { margin-top: 1px; background: transparent url(image/images/dark-tab.png) repeat-x top; }
.tab-strip a.tab-left {
background: none;
height: 24px;
line-height: 24px;
-moz-border-radius: 6px 6px 0 0;
border-radius: 6px 6px 0 0;
-webkit-border-radius: 6px; /* Used multiple s mobile doesn't like this in shorthand */
-webkit-border-bottom-right-radius: 0;
-webkit-border-bottom-left-radius: 0;
}
.tab-strip a.tab-left:hover { background: transparent url(image/images/dark-tab-over.png) repeat-x top; }
.tab-table { border-width: 0 1px 1px 1px !important; border-color: #505050;}
```

```
.page-tabs{ border-width: 0 0 1px 0; }
.group-content {
padding: 5px 5px 3px 5px;
}
#pricing-table-wrapper {
margin-top: 10px;
}
```

4. Upload the file under Alternate CSS file.
5. Click **Accept**.
6. Navigate to the File Manager and upload the file tabs.zip to the **Images** folder.

Ordered Dates & Updated Workflow Dates

Updating the function

1. Navigate from: Process Definition > List Documents > List Attributes for Quote.
2. Click Add.
3. Create a new attribute named **Approval Date**. *The attribute type should be Date.*
4. On the General tab, set **Includes Time** to **No**.
5. On the Modify tab, change the row for **Revise to Revert** to **Default**.
6. On the Document Views tab, mark all steps and profiles as **Read-only**.
7. Add another new attribute named **Number of Days for Approval**. *The attribute type should be Integer.*
8. Navigate from: Process Definition > List Documents > List Library Functions for Quote.
9. Click on the Pricing function. *This location is based on the assumption your site was built on a Base Template that utilizes the Pricing library function. If that is not the case, navigate to the location that houses the main Pricing function that runs on Commerce.*
10. Update the function with the code below. *The code should not be placed in any loops and needs to be placed after the result variable has been declared, but before it is returned.*

```
// On Revise Set revision number and Submitted Date
if( lower(actionName) == "revise" ){
revisionNumber = revisionNumber_quote + 1;
result = result + "1~revisionNumber_quote~" + string(revisionNumber) + "|"
+ "1~submittedDate_quote~|"
+ "1~approvalDate_quote~|";
```

```
}
//Approval Submitted Date Stamp
if( lower(actionName) == "request approval" ){
submitDate = strtojavadate(substring(_system_date,0,10), "yyyy-MM-dd");
result = result + "1~submittedDate_quote~" + substring(datetostr(submitDate),0,10) + "|";
}
//Approval Date Stamp and Expiration Date calculations
if( lower(actionName) == "submit" ){
if(submittedDate_quote == "") {
submitDate = strtojavadate(substring(_system_date,0,10), "yyyy-MM-dd");
result = result + "1~submittedDate_quote~" + substring(datetostr(submitDate),0,10) + "|";
}
else {
submitDate = strtojavadate(substring(submittedDate_quote,0,10), "yyyy-MM-dd");
}
approvalDate = strtojavadate(substring(_system_date,0,10), "yyyy-MM-dd");
expireDate = adddays(approvalDate, adminExpirationTime_quote);
numApprovalDays = getdiffindays(approvalDate,submitDate);
result = result + "1~expirationDate_quote~" + substring(datetostr(expireDate),0,10) + "|"
+ "1~approvalDate_quote~" + substring(datetostr(approvalDate),0,10) + "|"
+ "1~numberOfDaysForApproval_quote~" + string(numApprovalDays) + "|";
}
```



Be sure to search the existing function for submitted_quote and approvalDate_quote to ensure no other code exists in the pricing function that is currently writing to those fields. If there is existing code, it should be replaced with this code.

NOTES



Alternate Stylesheet CSS

RELATED TOPICS

BASE TEMPLATE APPROVAL HISTORY

ADMINISTRATION

Adding a new Approval History

1. Navigate from: Process Definition > Printer Friendly/History XSL Views.
2. Click **Add**.
3. Create a new XSL view named **Approval History**.
4. Add the XSL file `ApprovalHistory.xsl`. *This file is attached below.*
5. Navigate from: Process Definition > List Documents > List Actions for Quote.
6. Click **Add**.
7. Create a new Display History action named **View Approval History**.
8. On the **XSL Views** tab, select the XSL views for Approval History that you just created.
9. On the **Document Views** tab, select the visibility desired for the action.



10. On the **General Tab**, select the Browse button for the Action icon.
11. Select the Hourglass Icon from the icons folder in the **File Manager**.
12. Click **Save and Edit Layout**.
13. Add the new Approval History action onto the Commerce layout where you want it displayed. *The recommendation is on the bottom center of the Approvals tab.*

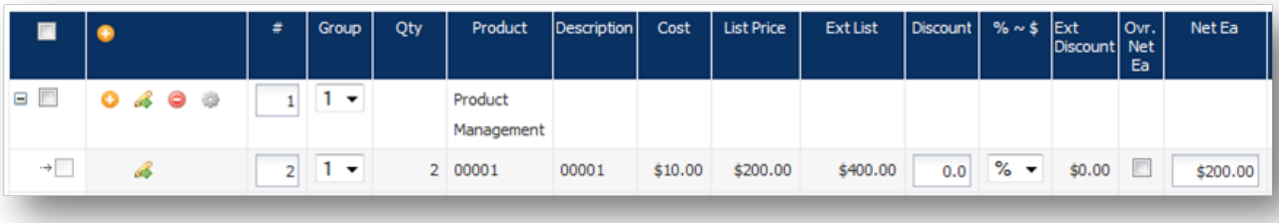
NOTES

Approval History XSL

RELATED TOPICS

BASE TEMPLATE LINE LEVEL

OVERVIEW



		#	Group	Qty	Product	Description	Cost	List Price	Ext List	Discount	% ~ $	Ext Discount	Ovr. Net Ea	Net Ea
		1	1		Product Management									
		2	1	2	00001	00001	$10.00	$200.00	$400.00	0.0	%	$0.00		$200.00

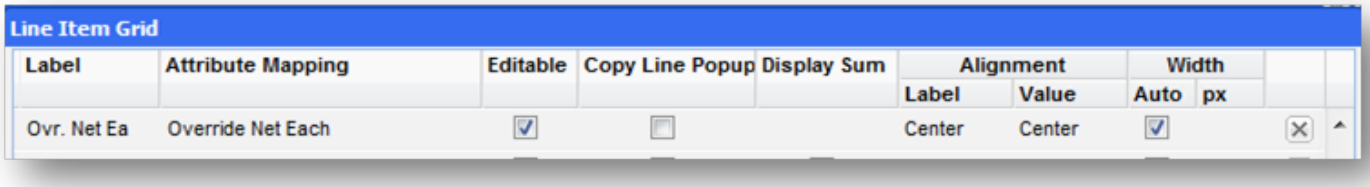
ADMINISTRATION

Line Level Goal Seek

The purpose of this is to allow sales users to specify the **Net Price** and reverse calculate the **Discount off of the List Price**.

Setting up Line Level Goal Seek

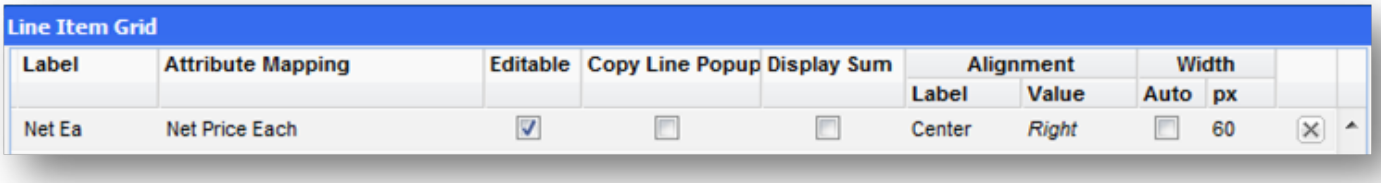
1. Add the Boolean attribute **Override Net Each (overrideNetEach_line)**.
2. Set the default to **False**.
3. Add a new column **(Ovr..Net Ea)** to the **Line Item Grid Layout**. *Place it after Ext. Discount.*



Line Item Grid

Label	Attribute Mapping	Editable	Copy Line Popup	Display Sum	Alignment		Width	
					Label	Value	Auto	px
Ovr. Net Ea	Override Net Each	☑	☐		Center	Center	☑	

4. Make the **Net Ea** columns **Editable**.



Line Item Grid

Label	Attribute Mapping	Editable	Copy Line Popup	Display Sum	Alignment		Width	
					Label	Value	Auto	px
Net Ea	Net Price Each	☑	☐	☐	Center	*Right*	☐	60

5. Modify the **Commerce Library Pricing Function**. *Do the following:*

   - Add the attribute **Override Neat Each (overrideNetEach_line)**.
   - Initialize the Variable.

```
overrideNetPriceDict = dict("float"); //key:document number
```

   - Add value into dictionary for both parts and models (this code is added in two places) when a **Net Price** has been specified by the user.

```
if(line.overrideNetEach_line){
put(overrideNetPriceDict, line_document_number, line.netPriceEach_line);
}
```

   - Set the temporary variable **Net Price** if override is **True** for Goal Seek. *This code is added in two places, both for model and parts code.*

```
//Sets Net Price if Override (Goal Seek)
if(containsKey(overrideNetPriceDict, eachDocNum)){
netPriceEach = get(overrideNetPriceDict, eachDocNum);
discountAmtOverd = listPrice-netPriceEach;
if(discountType == "%"){
if(listPirce<>0){
discount = (discountAmtOver/listPrice) * hundredPercent;
}
}
else{
discount = discountAmtOvrd;
}
}
```



The incorrect use and placement of this code could lead to incorrect pricing, errors and undesirable results. Ensure there is a full understanding of this code and the site where this will be placed before proceeding.

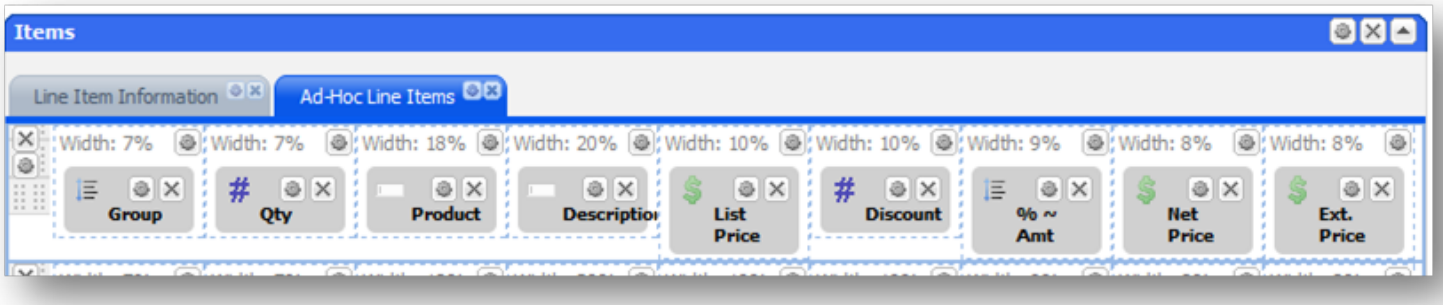
Ad-Hoc Items

1. Add the following attributes:

   - **Group (miscChargeGroupX_quote, where X is 1-6):** This is a menu type attribute and should have Options 1-6. The number of attributes created should match the number of Ad-Hoc items. In this example there are 6.

- **List Price (miscChargeListX_quote, where X is 1-6):** This is a currency attribute. The number of attributes created should match the number of Ad-Hoc items. In this example there are 6.
- **Discount (miscChargeDiscountX_quote, where X is 1-6):** This is a float attribute. Select Triggers Auto Update option. The number of attributes created should match the number of Ad-Hoc items. In this example there are 6.
- **Discount Percent (miscChargeDiscountPercentX_quote, where X is 1-6):** This is a float attribute. The number of attributes created should match the number of Ad-Hoc items. In this example there are 6.
- **% ~ Amt (miscChargeDiscountTypeX_quote), where 1-6):** This is a menu type attribute that should have the options % and Amt. Select Triggers Auto Update option. The number of attributes created should match the number of Ad-Hoc items. In this example there are 6.

2. Add the attributes **Group**, **List Price**, **Discount** and **%~Amt to the Layout Editor**.



3. Modify the **Commerce Pricing Library Function** by replacing the Ad-Hoc section with the code below (add the appropriate attributes to the rule).

```
// Miscellaneous Ad-hoc Charges

// Calculates the discount percent

discMiscCharge1 = miscChargeDiscount1_quote; discMiscCharge2 = miscChargeDiscount2_quote;

discMiscCharge3 = miscChargeDiscount3_quote; discMiscCharge4 = miscChargeDiscount4_quote;

discMiscCharge5 = miscChargeDiscount5_quote; discMiscCharge6 = miscChargeDiscount6_quote;

if(miscChargeDiscountType1_quote == "Amt"){ discMiscCharge1 = (miscChargeDiscount1_quote / miscChargeList1_quote)
* hundredPercent;}

if(miscChargeDiscountType2_quote == "Amt"){ discMiscCharge2 = (miscChargeDiscount2_quote / miscChargeList2_quote)
* hundredPercent;}

if(miscChargeDiscountType3_quote == "Amt"){ discMiscCharge3 = (miscChargeDiscount3_quote / miscChargeList3_quote)
* hundredPercent;}

if(miscChargeDiscountType4_quote == "Amt"){ discMiscCharge4 = (miscChargeDiscount4_quote / miscChargeList4_quote)
* hundredPercent;}

if(miscChargeDiscountType5_quote == "Amt"){ discMiscCharge5 = (miscChargeDiscount5_quote / miscChargeList5_quote)
* hundredPercent;}

if(miscChargeDiscountType6_quote == "Amt"){ discMiscCharge6 = (miscChargeDiscount6_quote / miscChargeList6_quote)
* hundredPercent;}
```

```
discMiscCharge1 = round(discMiscCharge1, numDecimal);

discMiscCharge2 = round(discMiscCharge2, numDecimal);

discMiscCharge3 = round(discMiscCharge3, numDecimal);

discMiscCharge4 = round(discMiscCharge4, numDecimal);

discMiscCharge5 = round(discMiscCharge5, numDecimal);

discMiscCharge6 = round(discMiscCharge6, numDecimal);

// Net Misc Charges. Net Charges are calculated by taking

netMiscCharge1 = miscChargeList1_quote - (miscChargeList1_quote * round((discMiscCharge1 / hundredPercent),
numDecimal));

netMiscCharge2 = miscChargeList2_quote - (miscChargeList2_quote * round((discMiscCharge2 / hundredPercent),
numDecimal));

netMiscCharge3 = miscChargeList3_quote - (miscChargeList3_quote * round((discMiscCharge3 / hundredPercent),
numDecimal));

netMiscCharge4 = miscChargeList4_quote - (miscChargeList4_quote * round((discMiscCharge4 / hundredPercent),
numDecimal));

netMiscCharge5 = miscChargeList5_quote - (miscChargeList5_quote * round((discMiscCharge5 / hundredPercent),
numDecimal));

netMiscCharge6 = miscChargeList6_quote - (miscChargeList6_quote * round((discMiscCharge6 / hundredPercent),
numDecimal));

// Extended Misc Charges. Extended Charges are calculated by taking Net Charge * Quantity

extMiscCharge1 = netMiscCharge1 * miscChargeQty1_quote;

extMiscCharge2 = netMiscCharge2 * miscChargeQty2_quote;

extMiscCharge3 = netMiscCharge3 * miscChargeQty3_quote;

extMiscCharge4 = netMiscCharge4 * miscChargeQty4_quote;

extMiscCharge5 = netMiscCharge5 * miscChargeQty5_quote;

extMiscCharge6 = netMiscCharge6 * miscChargeQty6_quote;

// Total Misc Charges.

totalMiscCharges = totalMiscCharges + extMiscCharge1 + extMiscCharge2 + extMiscCharge3 + extMiscCharge4 + extMis-
cCharge5 + extMiscCharge6;

subtotalPreQuoteDiscounts = totalNetPrice + totalMiscCharges;

// Adds Misc Items to Custom Group Subtotals

if(miscChargeGroup1_quote <> "" ){

customGroupSubtotal = get(customGroupSubtotalDict, miscChargeGroup1_quote);

customGroupSubtotal = customGroupSubtotal + extMiscCharge1;

put(customGroupSubtotalDict, miscChargeGroup1_quote, customGroupSubtotal);

}

else {

ungroupSubtotal = ungroupSubtotal + extMiscCharge1;

}

if(miscChargeGroup2_quote <> "" ){

customGroupSubtotal = get(customGroupSubtotalDict, miscChargeGroup2_quote);

customGroupSubtotal = customGroupSubtotal + extMiscCharge2;
```

```
put(customGroupSubtotalDict, miscChargeGroup2_quote, customGroupSubtotal);
}
else {
ungroupSubtotal = ungroupSubtotal + extMiscCharge2;
}
if(miscChargeGroup3_quote <> "" ){
customGroupSubtotal = get(customGroupSubtotalDict, miscChargeGroup3_quote);
customGroupSubtotal = customGroupSubtotal + extMiscCharge3;
put(customGroupSubtotalDict, miscChargeGroup3_quote, customGroupSubtotal);
}
else {
ungroupSubtotal = ungroupSubtotal + extMiscCharge3;
}
if(miscChargeGroup4_quote <> "" ){
customGroupSubtotal = get(customGroupSubtotalDict, miscChargeGroup4_quote);
customGroupSubtotal = customGroupSubtotal + extMiscCharge4;
put(customGroupSubtotalDict, miscChargeGroup4_quote, customGroupSubtotal);
}
else {
ungroupSubtotal = ungroupSubtotal + extMiscCharge4;
}
if(miscChargeGroup5_quote <> "" ){
customGroupSubtotal = get(customGroupSubtotalDict, miscChargeGroup5_quote);
customGroupSubtotal = customGroupSubtotal + extMiscCharge5;
put(customGroupSubtotalDict, miscChargeGroup5_quote, customGroupSubtotal);
}
else {
ungroupSubtotal = ungroupSubtotal + extMiscCharge5;
}
if(miscChargeGroup6_quote <> "" ){
customGroupSubtotal = get(customGroupSubtotalDict, miscChargeGroup6_quote);
customGroupSubtotal = customGroupSubtotal + extMiscCharge6;
put(customGroupSubtotalDict, miscChargeGroup6_quote, customGroupSubtotal);
}
else {
ungroupSubtotal = ungroupSubtotal + extMiscCharge6;
```

4. Add the following return string:

```
+ "1~miscExtCharge1_quote~" + string(extMiscCharge1) + "|"
+ "1~miscExtCharge2_quote~" + string(extMiscCharge2) + "|"
```

```
+ "1~miscExtCharge3_quote~" + string(extMiscCharge3) + "|"
+ "1~miscExtCharge4_quote~" + string(extMiscCharge4) + "|"
+ "1~miscExtCharge5_quote~" + string(extMiscCharge5) + "|"
+ "1~miscExtCharge6_quote~" + string(extMiscCharge6) + "|"
+ "1~miscCharge1_quote~" + string(netMiscCharge1) + "|"
+ "1~miscCharge2_quote~" + string(netMiscCharge2) + "|"
+ "1~miscCharge3_quote~" + string(netMiscCharge3) + "|"
+ "1~miscCharge4_quote~" + string(netMiscCharge4) + "|"
+ "1~miscCharge5_quote~" + string(netMiscCharge5) + "|"
+ "1~miscCharge6_quote~" + string(netMiscCharge6) + "|"
+ "1~miscChargeDiscountPercent1_quote~" + string(discMiscCharge1) + "|"
+ "1~miscChargeDiscountPercent2_quote~" + string(discMiscCharge2) + "|"
+ "1~miscChargeDiscountPercent3_quote~" + string(discMiscCharge3) + "|"
+ "1~miscChargeDiscountPercent4_quote~" + string(discMiscCharge4) + "|"
+ "1~miscChargeDiscountPercent5_quote~" + string(discMiscCharge5) + "|"
+ "1~miscChargeDiscountPercent6_quote~" + string(discMiscCharge6) + "|"
```

NOTES



Alternate Stylesheet

RELATED TOPICS

PERFORMANCE BEST PRACTICES

OVERVIEW

Maximizing Performance

- Turn on the configuration cache before go-live.
- Remove Print Statement - Remove all your print statement before go-live.
- Minimize number of loops, including nested and parallel loops.
- Don't use default functions at the line attribute level.
- Eradicate the use of Python.
- Do not use JavaScript.

ADMINISTRATION

Increasing Performance on a Site with Large Quotes

1. Use `dict("String")` instead of `dict("string[]")`. The speed on processing and storing `dict("string")` is faster than `dict("string[]")`. *The general idea is that the simple structure is faster.*
2. Eliminate **Revert to Default** on line item attributes if you have any for those actions. *They run X times, plus they invoke the line item advanced default script every time.*
3. Eliminate line item summation attributes. *They also run X times every update.*
4. If you need to loop through line items, store them into temporary variables; for example: string or float dictionary. After you store them, use the temporary variables than looping through line item documents collection.
5. If you have **Advanced Modify** on those actions, optimize the code:
    - Eliminate nested For... loops
    - Eliminate parallel For... loops
    - Eliminate table calls done in For... loops
    - When building the result string, use a temporary result string in loops in order to minimize the number of large string concatenations. For more information, see: http://vbforum.bigmachines.com/showthread.php?t=685.
6. Eliminate calls to **Global Functions**.
7. Eliminate or optimize quote level summation attributes.
8. Eliminate or optimize **HTML/Read-Only** attributes.
9. Eliminate **Revert to Default** for quote level attributes on those actions.

10. Ask OPS for the specs of the server and whether an upgrade is possible.
11. Delete any unused attributes.
12. Ensure that the first main doc always has `document number=1` instead of `res=res+_quote_doc_number+"~varname~"+value+"|"; use res=res+"|1~varname~"+value;`
13. Use the "Run Once For All Line Items" option on advanced line item default if you have any loops. Be sure to make sure the final results are correct. *If you do not select run once, the advanced default will run for each line item added. If you have many recommended items you are adding at once, this can cause a significant performance penalty.*

    Known Bug: "Run Once For All Line Items" is not respected when copying line items. If you make multiple copies of a configuration, the line item default will run for all copies being made.
14. Rather than calling the same data table multiple times with slightly different queries, get table data once, put it into an array, and process the data as you loop through the array. *The size of the data table and the number of queries will obviously influence the impact of this change.*
15. If calling large tables, look to reduce number of records by redesigning tables.
16. Minimize the number of RTE fields on line item attributes (no more than 5, as a general rule).
17. Minimize the number of line item attributes and the number of attributes displayed in the line item grid.
18. Do not put model/configurator-specific pricing in the pricing script. Instead, use the relevant configurator to calculate these.
19. Do not put large text area attributes on the page. *These will be submitted back to the server when an action is performed on the page, even if the action has "leave value unchanged" for the attribute.*
20. Regarding hiding rules, do not loop over line items to determine hiding behavior.

Configuration Performance Best Practices

- Minimize number of attributes per flow. *It will be faster to use multiple flows than one flow with many tabs.*
- Minimize array sets. *JavaScript and other factors can slow down configuration for arrays, especially in Internet Explorer.*
- Do not use image menus; these need to be loaded on the page. *The larger the images, and the greater the number of images, the more performance will suffer.*

Avoiding redundant lists

Having a list of items that repeats a word(s) in all the names is an all-too-common occurrence. Take advantage of labeling your groups more appropriately to avoid redundant content that distracts from the true distinct differences of the items.

- **Identify words that are used in every line of a group of labels.** Many groups of labels have one word that appears in all of them; it draws the eye away from what makes each line unique, and it creates clutter. *Remove this word and incorporate it into the group header.*
    - Using the word "Total" on every line of this list is unnecessary because they're all totals. Instead use a useful group label.

Good List	Bad List
TOTALS • Retail Price • Total Savings • New Price • Taxes • Net Price	• Total Retail Price • Total Savings • Total New Price • Total Taxes • Total Net Price

- **Eliminate redundant attribute choices.** It is common to have repeated words in the options listed for an attribute. Pinpoint the word being used in all of them and make each one look more unique. **Example:**

Redundant List
MEMORY • 2GB DDR SDRAM • 4GB DDR SDRAM • 6GB DDR SDRAM • 8GB DDR SDRAM

    - The longer, redundant aspects of the names do not need to be repeated. All of the items in this list are memory of type "DDR SDRAM". We could label this Attribute as "MEMORY (in GB of DDR SDRAM)" or something similar and our list could simply be numbers.

Hiding irrelevant content for different user types

- **Determine Hiding and Constraint Rules.** Avoid dumping all site features onto the screen by utilizing Hiding and Constraint rules. This allows you to only show applicable information to the specific user. *Ask yourself the following:*
    - *How many different end user types will there be?*
        - **Example:** My Tool Company consists of dealers and inside sales. There are three unique inside sales groups, each selling a different line of tools.
    - *How much information does each user type need to see?*
        - **Example:** The dealers of My Tool Company require all of the fields that we have included for Commerce, but the inside sales users only really need to collect about 40% of that information.

    - **Example:** The Config section has three tabs, one for each of the three lines of tools offered by My Tool Company. As mentioned before, there are three inside sales groups, and each group only sells from one line of tools.

- **Hide or Constrain Irrelevant Content.** Make the experience greater for all of your different types user by hiding or constraining the irrelevant content.
    - **Example:** An inside sales rep logs into BigMachines and only sees the Config options for the tools that she sells. When she enters Commerce, she has 60% less information to populate. At no point does this feel daunting or overwhelming because all the information presented is relevant to her specific role.

Using Boolean attributes when applicable

Boolean attributes are designed to handle true/false and yes/no choices. It is recommended that you use a Boolean attribute instead of a two-option single select menu. *It is considered best practice to be mindful of what you are asking the user:*

- **Example:** If you are asking the user whether they are a student...

DO	DON'T
Make a Boolean attribute that asks "Currently a Student"? *The user will know to select the box next to it.*	Make a Text Attribute with a drop-down menu or radio button list with the choices "Yes" and "No".

- **Example:** You are making a Boolean attribute to replace a text attribute you have called "Purchase History" with the following choices: "I have purchased items in the past" and "I am a new customer."
    - Change the label from "Purchase History" to "Have you purchased items from here before?" or "I have purchased items in the past" so that the user clearly knows what they are choosing to be True.
    - If the label is carefully reworded to avoid ambiguity, many two-option attributes can be turned into Booleans. *The user should be able to understand the option that they are selecting.*



It is generally recommended to set the default to "False".

RELATED TOPICS

VIDEO DEMONSTRATIONS

OVERVIEW

This section includes video demonstrations of features according to release. Click the links to watch; the video will open in a pop-up dialog.

ADMINISTRATION

Demos of BigMachines 2013 Autumn

- Mobile
- Secure File Manager
- Migration Rollback
- Help Site

Demos of BigMachines 2013 Spring

- User Management Enhancements
- Currency Enhancements
- Delta Data Table Uploads (Natural Keys)

Demos of BigMachines 2013 Winter

- Email Template Translations
- Formula Management
- Approval Sequences

Demos of BigMachines 12.2

- Customer Specific Pricing
- Simple Array Rules
- Formula Management
- Email Templates
- Simple Table Based Rules Array Scan

Demos of BigMachines 12.1

- Table Based Recommended Item Rules
- Boolean Attributes
- Pending Configurations
- Approval Skip Nodes
- Data Table Change Logs
- Localization

Demos of BigMachines 12

- Configuration Layout Editor
- Migration Management
- Commerce Rules

Demos of BigMachines 11.1

- AJAX Rules
- BMQL Transaction
- Document Migration
- Conditional Columns
- Price Book Enhancements

RELATED TOPICS

PERFORMANCE BEST PRACTICES

OVERVIEW

Maximizing Performance

- Turn on the configuration cache before go-live.
- Remove Print Statement - Remove all your print statement before go-live.
- Minimize number of loops, including nested and parallel loops.
- Don't use default functions at the line attribute level.
- Eradicate the use of Python.
- Do not use JavaScript.

ADMINISTRATION

Increasing Performance on a Site with Large Quotes

1. Use `dict("String")` instead of `dict("string[]")`. The speed on processing and storing `dict("string")` is faster than `dict("string[]")`. *The general idea is that the simple structure is faster.*
2. Eliminate **Revert to Default** on line item attributes if you have any for those actions. *They run X times, plus they invoke the line item advanced default script every time.*
3. Eliminate line item summation attributes. *They also run X times every update.*
4. If you need to loop through line items, store them into temporary variables; for example: string or float dictionary. After you store them, use the temporary variables than looping through line item documents collection.
5. If you have **Advanced Modify** on those actions, optimize the code:
    - Eliminate nested For... loops
    - Eliminate parallel For... loops
    - Eliminate table calls done in For... loops
    - When building the result string, use a temporary result string in loops in order to minimize the number of large string concatenations. For more information, see: http://vbforum.bigmachines.com/showthread.php?t=685.
6. Eliminate calls to **Global Functions**.
7. Eliminate or optimize quote level summation attributes.
8. Eliminate or optimize **HTML/Read-Only** attributes.
9. Eliminate **Revert to Default** for quote level attributes on those actions.

10. Ask OPS for the specs of the server and whether an upgrade is possible.
11. Delete any unused attributes.
12. Ensure that the first main doc always has `document number=1` instead of `res=res+_quote_doc_number+"~varname~"+value+"|"; use res=res+"|1~varname~"+value;`
13. Use the "Run Once For All Line Items" option on advanced line item default if you have any loops. Be sure to make sure the final results are correct. *If you do not select run once, the advanced default will run for each line item added. If you have many recommended items you are adding at once, this can cause a significant performance penalty.*

    Known Bug: "Run Once For All Line Items" is not respected when copying line items. If you make multiple copies of a configuration, the line item default will run for all copies being made.
14. Rather than calling the same data table multiple times with slightly different queries, get table data once, put it into an array, and process the data as you loop through the array. *The size of the data table and the number of queries will obviously influence the impact of this change.*
15. If calling large tables, look to reduce number of records by redesigning tables.
16. Minimize the number of RTE fields on line item attributes (no more than 5, as a general rule).
17. Minimize the number of line item attributes and the number of attributes displayed in the line item grid.
18. Do not put model/configurator-specific pricing in the pricing script. Instead, use the relevant configurator to calculate these.
19. Do not put large text area attributes on the page. *These will be submitted back to the server when an action is performed on the page, even if the action has "leave value unchanged" for the attribute.*
20. Regarding hiding rules, do not loop over line items to determine hiding behavior.

Configuration Performance Best Practices

- Minimize number of attributes per flow. *It will be faster to use multiple flows than one flow with many tabs.*
- Minimize array sets. *JavaScript and other factors can slow down configuration for arrays, especially in Internet Explorer.*
- Do not use image menus; these need to be loaded on the page. *The larger the images, and the greater the number of images, the more performance will suffer.*

Avoiding redundant lists

Having a list of items that repeats a word(s) in all the names is an all-too-common occurrence. Take advantage of labeling your groups more appropriately to avoid redundant content that distracts from the true distinct differences of the items.

- **Identify words that are used in every line of a group of labels.** Many groups of labels have one word that appears in all of them; it draws the eye away from what makes each line unique, and it creates clutter. *Remove this word and incorporate it into the group header.*
    - Using the word "Total" on every line of this list is unnecessary because they're all totals. Instead use a useful group label.

Good List	Bad List
TOTALS • Retail Price • Total Savings • New Price • Taxes • Net Price	• Total Retail Price • Total Savings • Total New Price • Total Taxes • Total Net Price

- **Eliminate redundant attribute choices.** It is common to have repeated words in the options listed for an attribute. Pinpoint the word being used in all of them and make each one look more unique. **Example:**

Redundant List
MEMORY • 2GB DDR SDRAM • 4GB DDR SDRAM • 6GB DDR SDRAM • 8GB DDR SDRAM

    - The longer, redundant aspects of the names do not need to be repeated. All of the items in this list are memory of type "DDR SDRAM". We could label this Attribute as "MEMORY (in GB of DDR SDRAM)" or something similar and our list could simply be numbers.

Hiding irrelevant content for different user types

- **Determine Hiding and Constraint Rules.** Avoid dumping all site features onto the screen by utilizing Hiding and Constraint rules. This allows you to only show applicable information to the specific user. *Ask yourself the following:*
    - *How many different end user types will there be?*
        - **Example:** My Tool Company consists of dealers and inside sales. There are three unique inside sales groups, each selling a different line of tools.
    - *How much information does each user type need to see?*
        - **Example:** The dealers of My Tool Company require all of the fields that we have included for Commerce, but the inside sales users only really need to collect about 40% of that information.

    - **Example:** The Config section has three tabs, one for each of the three lines of tools offered by My Tool Company. As mentioned before, there are three inside sales groups, and each group only sells from one line of tools.

- **Hide or Constrain Irrelevant Content.** Make the experience greater for all of your different types user by hiding or constraining the irrelevant content.
    - **Example:** An inside sales rep logs into BigMachines and only sees the Config options for the tools that she sells. When she enters Commerce, she has 60% less information to populate. At no point does this feel daunting or overwhelming because all the information presented is relevant to her specific role.

Using Boolean attributes when applicable

Boolean attributes are designed to handle true/false and yes/no choices. It is recommended that you use a Boolean attribute instead of a two-option single select menu. *It is considered best practice to be mindful of what you are asking the user:*

- **Example:** If you are asking the user whether they are a student...

DO	DON'T
Make a Boolean attribute that asks "Currently a Student"? *The user will know to select the box next to it.*	Make a Text Attribute with a drop-down menu or radio button list with the choices "Yes" and "No".

- **Example:** You are making a Boolean attribute to replace a text attribute you have called "Purchase History" with the following choices: "I have purchased items in the past" and "I am a new customer."
    - Change the label from "Purchase History" to "Have you purchased items from here before?" or "I have purchased items in the past" so that the user clearly knows what they are choosing to be True.
    - If the label is carefully reworded to avoid ambiguity, many two-option attributes can be turned into Booleans. *The user should be able to understand the option that they are selecting.*



It is generally recommended to set the default to "False".

RELATED TOPICS

IDENTIFYING THE SITE NAME

OVERVIEW

From within a script of HTML document, it may be necessary to dynamically identify the current site's identity. There are a lot of different ways to accomplish this task, which vary based on the context and location within a BigMachines application.

Determining the identity of the current site is a critical step in creating portable, safe applications that can be copied across different testing, staging, and production instances. Any time a resource is being referenced, externally or internally, make sure that a migration or site copy will not cause issues.

Some common areas where this comes into play

- Accessing files or images stored in the File Manager
- Connecting to an external integrated service
- Sending a call to the BigMachine's SOAP Services
- Document Engine XSL Snippets
- Header/Footer images, CSS, and JavaScript

ADMINISTRATION

Relative vs Absolute Links

A relative link doesn't include the hostname, and refers to a resource based on it's starting position. Relative Path Examples:

```
<a href="/path/to/resource.html"></a>
<a href="resource.html"></a>
<a href="../resource.html"></a>
```

There are some instances where using a relative path is enough. The path will automatically point to the appropriate resource, regardless of the environment.

The challenge with linking to the file manager is that the name of the site appears in two locations:

`http://*SITENAME*.bigmachines.com/bmfsweb/*SITENAME*/image/`

This can make it difficult to find link to the resource without having the site name available. Our strategy for finding the site name will vary based on where we are in BigMachines.

In Commerce BML

Use **_system_supplier_company_name** to identify the current site. This is a variable that stores the hostname of the current site, and is the same name that appears in the relative path to the file manager.

```
sitename = lower(_system_supplier_company_name);
filemanager_path = "/bmfsweb/"+sitename+"/image/";
```



The path to the file manager is case sensitive. Be sure to account for this by using "lower()" in your function.

In Configuration BML

You can use BMQL in combination with the commerce method to expose the site name in configuration. Assuming that the Quote Variable Name of the Commerce Document is "quote_process" the script would look like this:

```
path = "";
records = BMQL("select filemanager_path from commerce.quote_process");
for item in records {
path = get(item, "filemanager_path");
}
return path;
```

Doc Engine XSL Snippets

There are some paramaters available in the snippets that provide site-specific information:

- **$XSL_COMPANYNAME_PARAM**: Returns the site name.
- **$XSL_URL_PARAM:** Returns the site URL, including the protocol.(http://sitename.bigmachines.com/)

```
<!-- Example XSL Snippet-->
<xsl:variable name="image_url">
<xsl:value-of select="concat($XSL_URL_PARAM, '/bmfsweb/',$XSL_COMPANYNAME_PARAM, '/image/my_folder/my_image.gif"/>
</xsl:variable>
```

Display XSL Templates

When modifying the Home Page XSL, Parts Search XSL, or other XSL display templates, the site name is available under the "page_data" element. Here's how to select it in the Home Page. Notice that we translate it to lower case for the file manager link:

```
<xsl:variable name="sitename" select="translate(/home_page/page_data/host_company, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz')"/>
<xsl:variable name="image_url" select="concat('/bmfsweb/',$sitename,'/image/my_folder/my_image.jpg')"/>
```

The $BASE_PATH$ Parameter

$BASE_PATH$ is a variable supplied by BigMachines, that will resolve to the full relative link of the file manager. If you are adding a file to the header or footer, you can use the variable like so:

```
<img src="$BASE_PATH$/my_folder/my_image.jpg">
```

$BASE_PATH$ is available in the following areas:

- Header
- Footer
- Custom Home Page
- Commerce HTML/Read-Only Attribute
- Configuration HTML/Read-Only Attribute
- Configuration Flow Custom Templates

Home Page Alt JS File

From within the Home Page Alt JS file, you will have access to the global variable _BM_HOST_COMPANY, which refers to the current site name.

```
// outer function to avoid creating global variables
(function() {
var image_url = "/bmfsweb/"+_BM_HOST_COMPANY+"/image/my_folder/my_image.jpg";
}());
```

From Within the File Manager

Some files that you link to within the File Manager can have references within themselves. CSS files, for instance, may have links to images.

In the case of CSS, you can use relative links within your file to find other files in the File Manager. As long as the folder structure stays consistent, this will work across sites.

For example, assume the following directory structure in the file manager:

- stylesheets/
    - style.css
- images/
    - photo.jpg

From within style.css, you can link to photo.jpg like so:

```
.class-foo {
background-image: url(../images/photo.jpg);
}
```

TROUBLESHOOTING

It can be difficult to recognize that a static reference has been broken, especially in the case where it points to a similar resource on a BigMachines. If there is a broken link, or a visible error, then this could be good news. It will point directly to the source of the problem. Using the global script search to search for the site name is another quick way to find hard-coded references. **Firebug has some features that can be helpful:**

- Check the "Net" panel for any calls to an external resource.
- Use the Multi-file search to check for the name of a test, production, or staging instance.
- Try it from a commerce document, from the home page, and from your configurators.

As an extreme measure, Ops can bring down a test site for a testing period. During this period a thorough screen should be run over the buyside, looking for broken links and errors in functionality.

NOTES



Having links that point to unexpected locations is a maintenance liability and security risk.

RELATED TOPICS

SOURCE CONTROL AND VERSIONING

OVERVIEW

What

Source control refers to the process of tracking changes across types of "source code" that can be used to generate the application. Versioning is a more general term that refers to keeping snapshot revisions of each major version of a particular resource. For more information, including a reference to many of the terms you will see here, please see http://en.wikipedia.org/wiki/Revision_control.

Why

Keeping track of changes, and why they were made, is important for any project. **It supports:**

- The original work creator, to know what changes were related to a given piece of functionality.
- Other developers in a collaborative work, who may be working on different parts of the same piece of code, to avoid conflicts and overwriting each other.
- Other users who may have to pick up the original work and extend/maintain it.

Where

You should use source control anywhere you have code. In addition, it is usually a good idea to keep all major versions of any other types of documents (e.g. datasheets, contracts), especially if they are published or distributed in some form. BigMachines does not have versioning built in, but there are a variety of tools that you can use outside the system.

How

The first step is to get a good version control system. There are a number of these, many of them free. If you have a standard within your organization, that is usually the best one to use. If you do not have a standard that your BigMachines admins are already using, here are a few that you can try:

- **Git:** http://git-scm.com/ - Distributed Version Control, very good Windows client at http://code.google.com/p/tortoisegit/
- **Mercurial:** http://mercurial.selenic.com/ - Distributed Version Control, very good Windows client at http://tortoisehg.bitbucket.org/
- **Subversion:** http://subversion.tigris.org/ - Centralized Version Control, very good Windows client at http://tortoisesvn.net/
- **Dropbox:** http://www.dropbox.com or **Box:** http://www.box.com are good options for document snapshots, with the added benefit that you can easily use those services to share documents with other users.

Next, you need some policies and guidelines. This is really dependent on your development procedures, but here a few practices that we recommend:

- As mentioned above, version control everything that you can. Similar to backup, if you neglect to version control of one script or document, that will probably be the one that you will want history of at one point. In BigMachines, you would definitely want to control at least the following: BML, XSL, CSS, Javascript. You may also want to include things like Data Tables (which are easily exported as CSV and can be versioned very efficiently).

- Check in often. The smaller the individual changesets are, the easier it will be to determine what the purpose of each one is. A good rule of thumb is, commit your changes whenever you click the “Apply” or “Update” button to save your changes in BigMachines. One of the benefits of a distributed version control system like Mercurial is that you can do this without impacting other users, since commits are local by default.

- Pay especially close attention to scripts that more than one user are working on (common configuration rules or commerce pricing functions, for example). If you are a developer in one of those scripts, and you know that other users are working on it, make sure to GET the latest version before making any modifications of your own, rather than reusing a version you have saved offline.

- Make sure your system automatically tags changes with the user that makes the change, or make it a matter of policy to do so. Usually, when you want to know what a piece of code does, the best person to ask is the original developer, and this makes it much easier to know who that was.

- If your system allows it, add a COMMENT to each checkin/commit in order to track what PURPOSE the changes are meant to address. This would be a good place to put a reference to a case, requirement, or design feature, that the code implements. Internally, we use salesforce.com Cases to track almost everything we build, so we have a standard field for case number that automatically links to the Case, but a simple text comment works as well.

- Ensure that you can revert to earlier versions, and know how to do so. This is especially important in a crisis situation, when you want to be able to recover quickly and efficiently, and often a revert is the best way to accomplish this.



Ensure that everyone in your organization that touches the code follows the source control policy, if at all possible. This is true for all policies that you make, but it is especially important for source control because that system is the source of record for all of the changes that you make, and may save you from having to recreate a lot of lost work when somebody accidentally overwrites what you did.

Other Solution Components

Some elements (e.g. Attributes, Simple Rules) are not easy to put under version control because they are not expressed as text. In practice this rarely creates an issue, since these components are fairly simple to modify/delete/recreate, but if you are concerned about this, there are a few steps you can take:

- Keep a changelog of all changes made, including Attributes and Rules, with enough detail to be able to recreate them if needed. The Administration Logs (Admin → Developer Tools → Administration Logs) can be useful as a starting point, which you can embellish as necessary.

- Periodically download and store “snapshots” of the full Configuration/Commerce Process using the Bulk Upload/Download feature.

RELATED TOPICS

IMAGE BEST PRACTICES

OVERVIEW

It is recommended to format images to the appropriate size prior to uploading them into the File Manager. *This holds true for all images, whether they are being used in the image menus, as icons, or elsewhere.* Keep the following in mind:

- Be consistent with the image sizes.
- Icons will be scaled up/down based on how big the group headers are.
- The recommended maximum image size is 100px by 100px.

ADMINISTRATION

Using images in the Header & Footer

- Header images can be any size, but in most cases should be less than 200 pixels in height. *A large header can require the user to scroll unnecessarily to access site content.*
- Footer images should be avoided. *They take up page real-estate and can result in unnecessary scrolling.*

> For more information on Headers and Footers, see Header & Footer.

Using images in Config

- Easy to recognize Image Menus are the most useful. *Use the same image size within the Image Menu for continuity.*
- A simple object on a white background is most effective. *Avoid using a photograph with scenery/details that may be confusing or distract from the focus of the image.* **Example:**
  - **Clear Focus:** Image of a sliced ham on a white background makes it clear that the product is "ham".



  - **Unclear Focus:** Image of sliced ham on a plate with some vegetables, side dishes, and candles. This is confusing and does not clearly portray the product.



- Using product logos encourages brand recognition and are easily identifiable.
- Display the selection's details above or below the image for clarity. *Unless the name appears in the image, or the information is directly obvious from the image.*
- Avoid using a large number of selections in the Image Menu.
    - Too many images in a Grid Image Menu can clutter your page and cause confusion.
    - Too many images in a drop-down Image Menu can cause unnecessary scrolling.
- Be mindful of image size.
    - An image that is too large increases page load time, and if overused, clutters your page.
    - An image that is too small can be difficult to see, and details can be lost.

Using images in Group Labels

- Useful for branding, design consistency, and visual appeal.
- Can help to differentiate Groups. *This makes the page feel more organized and structured.*
- Be mindful of image size.

Using images in the Commerce Process

- Images can provide a visual cue of where the user is in the process. **Example:** a chain of images depicting "Pending Approval > Under Review > Finalized".
    - Current stage is highlighted, while future stages are grayed out.
    - Quickly communicates what stage the user is currently in, and what stages are left.
- This gives the user a sense of progress and adds visual appeal.

Using images in Doc Engine

- Using the company logo in the Header/Footer area provides branding continuity.
- Images can be used to dynamically insert signatures during the Approval Process.
- Logos or Product Images from the current order can add visual appeal and immediate recognition of the user's selections.

- A watermark image can be applied to documents on PDF outputs.
    - Can be used to indicate what stage of the process an order is in. **Example:** "Sample", "Pending Approval", "Submitted"



Background images (watermarks) are not supported in RTF format.

- JPEG and non-transparent GIF files are the only acceptable types of image formats.
    - PDF format supports JPEG and non-transparent GIF files.
    - RTF format only supports JPEG format.
- It is preferable that all images be stored in the File Manager. For more information on secure File Manager images, see File Manager.
- Image dimensions should not exceed the editor width or table cell width, depending on where you are inserting the image.



For more information on Doc Engine, see Document Engine Overview.

Using images in the File Manager

- Upload your images to the File Manager using the actual size and resolution with which they will be displayed.
- Do not let your images be re-sized or stretched.



Images can be secured within the File Manager, for more information, see File Manager.

Replacing default icons

1. It is considered best practice to replace a file in File Manager only in situations where it is absolutely necessary. *Follow these steps to determine the best approach:*
2. If you wish to simply hide the icon, follow best practices for hiding elements within the CSS. For more information, see Styling Best Practices.
3. Confirm that the icon is not used anywhere else in the site (where you wouldn't want it changed).

4. Use Firebug's inspector tool to uncover whether the icon image is defined in the HTML or CSS. *This will determine how you can change the image.*
    - **HTML-defined icons:** If the image URL is located within the HTML, then the URL cannot be edited to display a different image. You must replace the file itself in File Manager.
        - Confirm that you wish to replace this image site-wide. *Check the rest of the site to see where else the image may be used.*
        - Create a copy of your old image before you place your replacement image in the File Manager.
    - **CSS-defined icons:** If you do not see the image URL in the HTML, check the Style tab in Firebug. You should find CSS that assigns the image URL via a "background" or "background-image" line. You can replace this image with a new file and altered CSS.
        - Add your replacement image to the File Manager
        - Using the new image's URL, manually override the background-image value in the stylesheet.
        - You have flexibility with the specificity of your icon replacement because it is not site-wide. You can replace the icon only in certain places and leave it alone elsewhere.



Maintain a record of any files you have replaced or added in the File Manager. *This helps ensure a smooth migration from test to production.*



Your replacement icons should be created or edited by a professional and shouldn't be an image that is not owned or copyrighted by any entity that has not given permission. *It should be created or edited in a program like Photoshop that preserves transparency (PNG is the recommended format for new files).*

Example: No existing image URL

1. Utilize Firebug to explore the HTML.
2. You find that there is no image URL. *Only the following code:*

```
<div class="action-icon action-icon-visible line-item-edit" actionid="4654394" title-
e="Edit"></div>

You can see in Firebug the following CSS code with the URL:

.line-item-edit {

background-image: url("/img/gxt/edit.png");

}
```

3. Obtain an approved image (preferably of the same height/width).
4. Uploaded this image to File Manager as "edit_new.png." For more information on secure files, see File Manager.
5. Enter the Alt or Commerce stylesheet and properly override the URL. *Example code:*

```
.action-icon-bar .line-item-edit {
background-image: url("/img/gxt/edit_new.png");
}
```

Example: Existing image URL

1. Your wish to use a blue plus icon instead of a green one in your Commerce Action button, "Add Line Items."
2. Utilize Firebug. *It shows you that the plus icon URL is written in the HTML itself:*

```
<img border="0" src="/bmfsweb/myClient/image/Icons/add.png" style="width: 16px; height: 16px;
top: -1px; left: -3px; position: absolute;">
```

3. Replace your "add.png" image in the File Manager. *A background image won't override the code.*
4. Make note for future migrations that this file needs to be replaced.

NOTES



Images can be secured within the File Manager, for more information, see File Manager.

Images can be added to the Mobile Home Page Punch-Ins using the image Browse/Upload function, located on the Admin Home Page. *The recommended maximum image width and height: 235 px by 150 px*. For more information on Mobile, see Mobile Overview.



Naming convention and file management best practices for images:
-All lower case
-No special characters
-Short, descriptive names
-Group images together in File Manager folders; organize according to BigMachines functional areas

RELATED TOPICS

APPROVAL SEQUENCES BEST PRACTICES

OVERVIEW

The Submit Action and related SubActions (Approve, Reject, Revise, Request Approval) provide a much cleaner user interface for managing approvals. *Potential source of confusion:* the action clicked on the user-side does not always trigger the same action in the system. Below is a table that describes the user clicked action and the conditions on which action is performed by the system.

User-clicked action & conditions

User-Side Action	Conditions	System-Side Action
Submit	No Approvals Required. No conditions are met on any of the Reason Trees.	**Submit**
Submit	1 or more Approvals Required. Conditions are met on at least one of the Reason Trees .	**Request Approval**
Approve (green check mark)	This is the final Approver across all Reason Trees. No further Approvals Required.	**Submit**
Approve (green check mark)	1 or more further Approvals Required. There is at least one more approval pending across all Reason Trees.	**Approve**
Revise	Always	**Revise**
Reject	Always	**Reject**

ADMINISTRATION

- **Reset reasons on "Revise" action**

The Revise action does not send out emails after evaluating the reason tree and finding new active reasons.

- **Don't reset reasons on "Reject" action**

When resetting reasons on reject, they are immediately re-evaluated and emails are sent to the appropriate approvers. Therefore the Approval Description will not contain the correct information upon rejection.

- **Don't transition on "Reject"**

The intended functionality is that the salesrep will revise the transaction if the quote is rejected.

- **Return at least one approver when using advanced approver functionality**
- The functionality uses the format "1~user~company|2~group~company"
- The "1" denotes that it is a user that you are designating as the approver.
- The "2" denotes that it is a group that you are designating as the approver.

- The "|" can only be added in if there are multiple approvers. If it is included at the end or beginning it will cause an error.
- When using the advanced approvers, you must return an approver, it can not be blank.

- **Adhere to one pending approval step when setting up transitions**

Adding in multiple steps increases the complexity of the build. This is definitely the case when trying to transition after more than one approval. Most use cases can be handled with having a variety of participant profiles.

- **Keep parent action "Submit" and sub-action "Submit for Approval" advanced modifications equivalent**

The javascript popup comment box is generated based on the parent's advanced modification. If the parent modification effects the reason tree's evaluation differently then the sub-action, this can lead to inconsistent results.

- **Hide "Submit" attributes on pending step**

This will result in less confusion for the user.

- **Approval trees don't control access to profiles**

Add each potential approval group to the Approver profile, either using Advanced Forwarding or static User Access Rights. When a non-Approver accesses a Quote, they will be able to view the Quote using their profile, but will not be able to Approve/Reject.

Summary of "Submit for Approval"

- User clicks parent action (Submit)
    - Line level modify tab is evaluated if line level changes have happened
    - Advanced modification for update line is applied if there are line level changes
    - Advanced validation for update line is applied if there are line level changes
    - Quote level modify tab is evaluated for the Submit action
    - Advanced modification for the Submit action is applied
    - Advanced validation for the Submit action is applied
    - Reason tree is evaluated and 'Submit' attributes are updated
    - If there is approval needed, i.e. at least one reason evaluated to true, Modification to the transaction is not saved and the user gets a popup to enter comments.

- If there were no reasons to evaluate the modification would be saved and the parent action's transitions would fire
- User enters in comments and clicks submit

- The child 'Submit for approval' action is then fired
- Line level modify tab is evaluated if line level changes have happened
- Advanced modification for update line is applied if there are line level changes
- Advanced validation for update line is applied if there are line level changes
- Quote level modify tab is evaluated for the 'Submit for approval' sub-action
- Advanced modification for the 'Submit for approval' sub-action is applied
- Advanced validation for the 'Submit for approval' sub-action is applied
- Reason tree is evaluated and 'Submit' attributes are updated

---

- Any approvers approve or reject

On "Approve" the following occurs:	On "Reject" the following occurs:
• Line level modify tab is evaluated if line level changes have happened • Advanced modification for update line is applied if there are line level changes • Advanced validation for update line is applied if there are line level changes • Quote level modify tab is evaluated for the sub-action • Advanced modification for the sub-action is applied • Advanced validation for the sub-action is applied • Reason tree is evaluated and 'Submit' attributes are updated	• Line level modify tab is evaluated if line level changes have happened • Advanced modification for update line is applied if there are line level changes • Advanced validation for update line is applied if there are line level changes • Quote level modify tab is evaluated for the sub-action • Advanced modification for the sub-action is applied • Advanced validation for the sub-action is applied • Any reset reasons are reset • Reason tree is evaluated and 'Submit' attributes are updated

---

- If the user clicks the 'Revise' Sub action it evaluates the same as a normal modify action with only one difference, any reasons that are checked to reset are reset and the reason tree is evaluated, but emails are not sent.

RELATED TOPICS

BML CODING BEST PRACTICES

OVERVIEW

The purpose of good coding style is to write code that is readable. Readable code is easier to debug, extend, and modify, thus increasing your productivity and that of anyone else who will read your code. It is your duty as a programmer to adhere to style best practices.

Types of coding styles:

Structural Coding Style

1. The Function Header should describe the entire script's salient properties.

- The name of the script or function (for global BML search)
- Script location within BMI
- Parameters
- Return type
- Include short script descriptions or cautions for future programmers who might edit your code
- The name of the script author, date, function editors' names, and summarizing comments
  Example:

```
Name Date Comments
-------------------------------------------------------------------------------------------
John Doe 1/1/19 Created Function
Jane Doe 5/1/19 Added additional managers to notifications
```

2. Major sections of your code should be clearly marked and contain a brief explanation of its purpose, who wrote the code, and the date the code was written or modified.

- Provide visual contrast if possible
- Organize like chapters in a book or specific tasks in an algorithm
- Aids other programmers in identifying where they are in your script, and the current section's purpose
- Ensures your script is well structured and reminds you of what you did in different areas of a very large BML script (e.g. Commerce Pricing)
- Aids in quickly skimming or scanning BML script without having to read through multiple, individual lines of comments

3. Secondary sections of your code should also be clearly marked and contain a brief explanation, if necessary.

- Some major sections of your BML script can become very complex; decompose your steps and provide explanations for readers

4. Group related lines of code together.

- The proximity of related lines of code communicates to a reader that those lines are related
- This helps a reader identify which lines of code correspond to which pieces of functionality, therefore aiding: debugging, amendment, and extension

5. Follow conventional indentation and bracketing practices (alignment).
    - Use the Tab key in Notepad++
    - Every immediately nested code block should be indented exactly one tab to the right with respect to its parent content
    - Curly brackets should always open at the end of a line of code
        - *No code of any kind should follow on the same line after the opening of a curly bracket (which denotes the start of a nested code block)*
    - Curly brackets should always close at the beginning and end of a line of code
    - No code should exist on the same line as a closing curly bracket (which denotes the end of a nested code block)
        - *The only exception for the curly bracket rules should be the inner array declaration of a hard-coded 2D array*

Inline Coding Style

1. The name and format of a BML local variable should immediately communicate its purpose, intended behavior, and usage.

    Local variables used as constants should:
    - CAPITALIZE all letters
    - Use underscores to separate words in the name of the constant
        - E.g. `ITEM_DELIM = "*@*@*";`

    Local variables used as variables should:
    - The first letter should be lower case
    - The first letter of each subsequent word in the name should be UPPER CASE
    - DO NOT use underscores to delimit separate words in the name
        - E.g. `totalListPrice = 0.0`

    In all cases, the name of the variable should describe itself:
    - `totalListPrice` is the sum of all line item list prices

- Variable names for Arrays should have the "List", "Arr", "Array" and/or "2D" suffix
    - E.g. `docNumList`, `upgradedPartsArray`, `partsData2DArray`

Variable names for Dictionaries should have the "Dict" suffix, and have a comment at the end of the declaration explaining the mapping between Key and Value:

- E.g. `sequenceNumDict = dict(string) //Key: Document Number, Value: Sequence Number`
- E.g. `relatedLicenseToDocNumDict = dict(float) //Key: relatedLicense, Value: Document Number`
- The exception to this rule is when iterating over Records in a RecordSet
    - E.g. `"for partRecord in partRecords {"`

Variable names for RecordSets should have the "Records" suffix:

- E.g. `partRecords`

Variable names for Booleans should have the "is" or "has" prefix (answered with a "yes/no" answer):

- Possible exceptions include standard Booleans such as "debug"
    - E.g. `isUpgrade`, `isMandatory`, `hasRelatedLicense`

2. There should never be more than one statement on one line *(i.e. never more than one semicolon ending a statement)*.

3. Print statements should be extricated from convoluted code and separated into a debug-flag controlled if block, and if it doesn't cause the code to become less readable.

4. Functions with a single argument should always enclose that argument in parentheses e.g. `not(hasLicensing)` rather than `not hasLicensing`.

5. Extremely long single-line statements, such as String concatenations, should be split across multiple lines.



The repetition of these naming conventions will aid a reader in determining the purpose of any variable without having to refer to additional context.

RELATED TOPICS

ARRAY BEST PRACTICES

OVERVIEW

Building Arrays

1. Configurable array sets must be created the proper way in order for the +/- add/remove buttons to appear or work .
    - Create the array set BEFORE visiting the “Set” tab of the array attribute
    - Ensure that the array control attribute is in the same group as the array set, and is not hidden.
    - This order will always work:
        a. Create the array control attribute
        b. Create the array set
        c. Create the array attributes and add to the set
2. If you created the wrong way you DO NOT need to delete your attribute and recreate. Here are the steps to fix an Array Set that is not showing the increment / decrement buttons:
    - Do the following for each array in the array set:
        a. Go to the Array Attribute on the admin side.
        b. Go to the Set Properties tab.
        c. Change the Array Set from the real one to the [DEFAULT] one.
        d. Change the Array Control Attribute to the blank entry.
        e. Click Apply. *important step*
        f. Change the Array Set back to the real one.
        g. Click Update.
    - Once you have done that for all arrays, the little icons should return.
3. It is possible to have one Size attribute control multiple arrays, but not recommended, since it will lead to confusion in maintenance.

Recommendation Rules and Arrays

- Set Type Recommendations will work for an array attribute only if none of the values of those attribute elements is changed by the user. Once changed, the system treats it as an overwrite of the attribute and will not perform a Set recommendation on any elements that get added afterward.

- If you recommend a value for a string array of "" (blank) or $BM_UNCHANGED_STR$ (leave value unchanged), even if the value is blank, you will still get the recommendation message ("Don't select any value…").

User Experience with Arrays

- There is quite a bit of BM behind-the-scenes Javascript associated with array sets. The presence of many array sets will affect performance, especially in IE.
- When you add a new element to an array using the +/- buttons, the value it takes is the value of the previous last element (the now penultimate element). Defaults are not respected in arrays.
- The recommendation text for an array attribute always shows the variable name of the attribute, rather than the display name, as for non-array attributes.
- Maximum number of rows in an array is 200.
- Multi-select menus are not possible in arrays.

Arrays & Table-based Rules

- Simple Table-based Rules operate per index; they will not operate across indexes.
- Attribute type limitations are consistent with existing simple Table-based Rules.
    - If the Table column is numeric, then the Array Attributes must be numeric

---

NOTES



Maximum Array size is 200.

RELATED TOPICS

STYLING BEST PRACTICES

OVERVIEW

Navigate from: Admin Home Page > Style and Templates

ADMINISTRATION

Proper formatting

- Use Notepad++ or a CSS editor to write your custom CSS files. *This promotes proper syntax and improves formatting.*
- Use the tab key for indentation.
- Curly brackets should open at the end of a line with a selector, and close at the beginning of the line following the last property and value.
- Properly comment and organize your CSS when you make additions or changes.
- Beware of special characters. *Some characters are reserved in HTML, XML, or XHTML because the browsers will mix them with tags. So if you need to use a special character use the HTML Entity.*
    - **Reserved characters:** < > & "
    - HTML Entities can be numeric or named; they all start with "&" and end with ";".
    - Use an entity name instead of a number; it's easier to remember. ***Note:*** *some browsers may not support all entity names.*
    - Click here for a list of HTML character entities.
- Use the hex code value instead of typing a color name. *This ensures continuity for branding purposes.*
    - When removing a selector color: either comment out the property & value OR delete the entire line.
    - Never leave a value blank after the #.

GOOD	BAD

`.bundle-list {`
`  border-color: #000000;`
`  border-style: solid;`
`}`
	`.bundle-list {`
	`  border-color: black;`
	`  border-style: solid;`
	`}`
	OR
	`.bundle-list {`
	`  border-color: #;`
	`  border-style: solid;`
	`}`

Class and ID naming conventions

- Always use lower case.
- Always begin with a letter. *CSS rules allow you to begin with an _underscore, but this is not a best practice and should be avoided.*
- Do not use spaces. Separate words 'with-a-dash'
- No special characters.
- Use a short, descriptive name.
    - Create descriptive names that describe the function, not the appearance of the section.
    - **Example:** Name an element located on the right-hand side of the page but will hold a list of bundle names, "'bundle-list" instead of "right-hand-column."
- **ID:** use the variable name when possible.

Proper HTML tag usage

- **Headings:** <h1> through <h6>. *Avoid using heading tags simply to bold text or format something differently*.
    - **h1:** Intended to be the page/main subject title (think of this as a book title). *Search engines often use h1 tags when looking at your site; overusing h1 tags can flag your site as spam.*
    - **h2:** Intended for section titles (think of these as chapter titles).
    - **h3-h6:** Sub-section titles; these are intended to be used in descending order.
- **Paragraphs:**
    - **
 tag:** Forces a line break within the text flow of a paragraph to continue your text on a new line. *Avoid using <br/> to end a paragraph or to change the length of your text lines.*
    - Avoid using   to add more space between elements. *It can result in awkward-looking HTML, which is confusing to edit later, and unexpected spacing in different browsers.*
    - Use a class name with the formatting found in your style sheets to achieve your desired spacing amount and type (e.g. indent, padding, first line indent, etc.).
- **Text emphasis**
    - CSS is the best way to format your text.
        - Assign a class to the text element (e.g. span, cell, group, grid). *That class can be used to format font-weight, font-face, size, color, alignment, etc.*
        - However, if you need to assign the styling within the HTML, use the correct tags.

    - **Bold text** Use . *<b> is an outdated tag; it works for now, but may not be supported in the future.*
    - **Italic text:** Use . *<i>is an outdated tag.; it works for now, but may not be supported in the future.*
- **Class and ID tags**
    - Always use lower case.
    - Always begin with a letter.
    - Use a short, descriptive name. *Create descriptive names that describe the function, not the appearance of the section.*
    - Don't use spaces: separate words "with-a-dash."
    - IDs should be used only for unique elements.
    - Classes can be used multiple times.
    - Always close your tags. *Failing to close a tag results in unwanted formatting, or "worst case scenario," can break your site (by interfering with JavaScripts or making critical elements inaccessible, invisible, or incorrectly displayed).*
- Always close your tags. *Failing to close a tag results in unwanted formatting, or "worst case scenario," can break your site (by interfering with JavaScripts or making critical elements inaccessible, invisible, or incorrectly displayed).*
- Use lower case markup.
- Avoid inline styles. *Instead, utilize CSS to style a class.*
    - Never use inline JavaScript.
- Validate your HTML.

Styles can be applied to Classes, ID's, and HTML elements

- Classes can be reused as needed throughout your site.
    - **Config:** classes can be applied to Tabs and Grids (within "Properties"). *If a class name is not specified, BigMachines will automatically assign a Grids name, based on default functionality.*
    - **CSS:** a class name always begins with a "." (e.g. '.group-list').
    - Avoid assigning either too many or too few classes to your elements.
- ID's are a unique styling: generally they're used to designate a special element (e.g. Body or Navigation bar) that are only used once per page.
    - **CSS:** ID names always begins with a "#" (e.g. '#nav-bar').

- Elements are the standard HTML components. **Examples:**
    - h1 = Heading 1
    - p = paragraph
    - td = table cell
- When styling an element, avoid applying a style across an entire site if that style is only intended for use in one section or a specific item.

Specificity in CSS

Specificity is a concept in CSS that determines how a conflict is resolved (e.g. competing CSS rules). The concept of specificity states that when two/more declarations apply to the same element, set the same property, and have equal importance/origin, the declaration with the most specific selector takes precedence.

- Be as specific as possible with your selector when writing a new CSS rule.
- When there is a conflict, utilize Firebug to discover which rule is conflicting with yours.
- Avoid using the `'!important'` tag to resolve conflicts that can be more easily resolved utilizing the principles of specificity.

Specificity example

1. Create the following rule in your Alt Stylesheet to display Configurator tab content 900 pixels wide:

```
tab-content {
width: 900px;
}
```

2. Check Firebug; something is wrong. After the CSS has been applied, the tab content remains at 1000 pixels:

```
.body-inner .tab-content {
width: 1000px;
}
```

3. After utilizing Firebug, you find that the above rule has a higher specificity value (which is why it takes precedence over the rule you've added to the Alt Stylesheet).

DO	DON'T

Make your rule more specific; adding another rule in the hierarchy gives your rule a higher specificity value.
Example:
`.body-inner .configuration .tab-content {`
`width: 900px;`
`}`
	Insert an "`!important`" tag. *This overloads your CSS.*

Calculating specificity

1. If one declaration is from a style attribute, rather than a rule with a selector (which is considered an inline style), it has the highest specificity. *If none of the declarations are inline, proceed to the next step.*
2. Count the ID selectors. The declaration with the highest count has the highest specificity. *If two/more have the same number of ID selectors, or if they don't have any ID selectors, proceed to the next step.*
3. Count the class selectors (e.g. `.test`), attribute selectors (e.g. `[type="submit"]`), and pseudo-classes (e.g. `:hover`). The declaration with the highest total has the highest specificity. *If two/more have the same total, or have totals of zero, proceed to the next step.*
4. Count the element type selectors (e.g. `div`) and pseudo-elements (e.g. `:first-letter`). The declaration with the highest total has the highest specificity.



According to CSS rules, if two/more selectors have the same specificity, then the latter specified rule takes precedence.

Hiding elements using CSS

Before hiding elements using the CSS, ask yourself the following questions:

- *Is the element part of an Attribute or something that can be hidden using Hiding Rules?* If yes, utilize BigMachines rules instead. Hiding an element with CSS doesn't remove it from the HTML (therefore, can still be accessed).
- *Can the element be removed through the Layout Editor or by changing the flow?* If, yes, do it that way.

Hiding elements

1. Open Firefox Firebug.
2. Use Firebug's inspect tool to highlight and select the div/element you want hidden. *This will direct you to its HTML.*
3. Ensure that your selection isn't too broad or too specific. *What is highlighted with Firebug is what will be hidden.*
4. Note the element's class and/or ID within the HTML. *An ID is preferred, since it should only apply to this element; classes require more attention as a class may be used by other elements within the site.*
5. Create a rule in the CSS. *Make the selector as specific as possible.*
6. Place the line, `"display: none;"`, within the CSS rule. *Comment the rule appropriately so others know what is being hidden.*
7. Check your results and site to ensure the correct elements have been hidden.

Example

Your site has the drop-down menu, titled, "Price Book". Because this element does not serve a purpose on your site, you would like to hide it. Utilize Firebug to locate it's class and create the following CSS rule (with proper commenting):

```
/*Hiding Price Book Element*/
.pricebook-container{
display:none;
}
```



Always consult your team members before hiding anything with CSS.

Adding/editing pre-existing CSS

In the process of implementing, upgrading, or maintaining a client site, you will likely find yourself needing to make a change to an existing stylesheet.

- Before replacing an existing stylesheet with your edited version, download and save a copy of the current stylesheet.
- Search the stylesheet before adding your change to see if a declaration with the same exact selector already exists.
    - *If the exact selector already exists w/in the CSS:* Place your change into the declaration rather than create a duplicate, conflicting one. Place a dated comment next to your changed line.
    - *If the particular selector/selector group doesn't already exist in the CSS:* Add new code. For more information, see Stylesheet Manager.
- When adding brand new code (e.g. new selector/selector group) to the stylesheet, place it in the appropriate section of related content. *If no such section exists or it is an isolated situation, place your code at the bottom and comment out a new section for it.*
- Place a comment above the selector with a clear explanation of what the CSS rule is accomplishing (include your initials & date if there are multiple people working on the CSS). *You can group related code together with a comment above the first rule and the word `"BEGIN"` followed by a comment below the last rule with the word `"END"`.*
- Maintain proper spacing between rules and comments.
    - Each rule should be spaced by (at least) one empty line from other rules.
    - Comments should be placed directly above the rules they reference.
    - Grouped comments (`"BEGIN"` and `"END"`) should have an empty line between the first and last rules they group.

Example

You want to change the width of class `.tab-content` to 1000px.

1. Search the stylesheet and find the existing `.tab-content` code.
2. Add the width declaration along with a comment, like this:

```
.tab-content{
width:1000px; /*Changed from 900 - JD, 4/21/12*/
}
```

3. If a declaration for `.tab-content` rule doesn't exist: add your new width declaration in the existing rule instead of making a duplicate selector.



Don't add a new rule for the selector .tab-content at the bottom of the stylesheet without checking to see if it exists. Adding a duplicate rule with a different value could create issues regarding which declaration takes precedence.



Always take care when adding new code. Use detailed comments, space appropriately, avoid redundancy.



Uncommented and disorganized code will make it more difficult for your co-workers to find and understand your changes. *It may also be difficult for you to remember the changes you have made later.*

External resources & design principles

Design principles

The following are concepts from *The Non-designer's Design Book* by Robin Williams.

- Begin with the end in mind: understand the end goal of the user and the site, so that your design is focused on the best way to reach that end goal.
- Never sacrifice legibility or usability.
- Contrast: if two items are not exactly the same make them very different.
- Repetition: repeat some aspect of the design throughout each page and throughout the site.

- Alignment: every item should have a visual connection with everything else on the page.
- Proximity: group related items together.

External resource: websites

Websites
UX Design
Yahoo UX
W3 Schools
W3C Markup Validation Service
A List Apart
Smashing Magazine
Eric A. Meyer's blog
Lynda.com *(paid subscription)*

External resources: books

Books
The Non-designer's Design Book - Robin Williams
Designing with Web Standards - Jeffrey Zeldman
Developing with Web Standards - John Allsopp
A Project Guide to UX Design: For user experience designers in the field or in the making - Ross Unger & Carolyn Chandler

---

NOTES



Before changing/adding a stylesheet: download the existing/default stylesheet to your local computer.
This is helpful if you need to revert changes or refer to the standard options.

Firebug is an extremely useful tool for locating element that you wish to style.



Document your changes and reasoning for said changes.

RELATED TOPICS

NAMING CONVENTIONS FOR RULES, ATTRIBUTES, AND VARIABLES

OVERVIEW

Engineers, architects, and admins have free reign over what they name rules, functions, documents, attributes, and internal variables. However, it is imperative that all of these aspects of an implementation are given clear, useful, and at least semi-standardized names. Consistent and clear naming supports maintenance and add-on activities. Well-named Rules and Attributes are easier to identify and code using well-named variables and functions is self-documenting.

Naming opportunities are ubiquitous across BigMachines:

- Rule naming applies to anywhere rules can be created; configuration rules are generally the focus.
- Functions are usually found in the form of utility or commerce libraries. *Though they may only be called in few places, it is still important that they have meaningful names.*
- Documents refer mainly to XSL integration files and DocEngine documents. *As sites become larger and more complex, it is imperative that administrators and engineers can quickly and easily identify what an integration file or DocEngine doc is for without having to open it up and study its components.*
- Attributes are extremely important because they are the link to the user and his or her experience with the product. *They are found in configurators and commerce.*
- Variables are contained in all of the above items. *If they are poorly named then maintenance and add-ons become difficult.*

ADMINISTRATION

Naming Convention Rules

The name of a configuration rule should give an administrator some insight as to what the rule does. Wherever possible, use names that evoke the business rules being implemented, using the terminology and language that the customer is familiar with. *Keep in mind that there are a number of attributes of rules stored as* `metadata` *so it is not necessary to recapitulate them in the naming (and it can clutter the UI if you do):*

- Rule Type
- Related Attributes
- Description
- Status (Active/Inactive/Internal)
- Model/Product Line/Product Family

Recommendation Rules

Recommendation Rules are generally used in order to “set” values based on other values, or “default” values conditionally based on the initial state (useful if multiple flows inherit a collection of attributes). **Examples:**

- “Defaults for NAM” might set a number of fields when the Quote Region is North America.
- “Set Voltage based on Power Supply” might implement a dependency relationship from Power Supply to Voltage (perhaps using a mapping table).
- “Require QuickStart Package for Users < 500” might force set Service Offering to QuickStart Package.



The important thing is to be consistent with the naming convention. Even if the site strays from the above recommended methods, consistent names make it easier for users to administer the site. If possible, decide early on in the implementation what the sort of naming convention will be used. If at any time you aren't sure, consult with the site's Solution Architect or lead engineer.

Constraint Rules

Constraint rules “limit”, “filter” or “remove” options that the user can select. **Examples:**

- “Filter Model by Product Line” might implement a hierarchical relationship between Product Line and Model.
- “Restrict Support Term by Support Type” might apply a table-based limit to the Support Term attribute which takes the Support Type attribute as input

---

Hiding Rules

Similar to other types of rules, name Hiding Rules based on the business rules they represent. *In most cases, the rule should describe “what” is hidden (and optionally “why”):*

- “Hide Advanced Rack Configuration” might hide an array set of customized options if an “Advanced Rack Configuration” option is set to “No”.
- “Hide Support Option if Subscription” might hide a “Support Option” attribute (or set) if a given item represents a Subscription (in which support is included).

Recommended Item Rules

Most sites have few enough rec item rules (some have just two or one) that it's okay to just refer to the output. It's generally understood that the rule is probably either always true, or fires when the user starts to provide the input that generates the recommended items. **Examples might include:**

- Spares – Optional
- Main Units – Mandatory
- Software – Mandatory



One thing that may be helpful is to include whether or not the rule type is Mandatory or Recommended. Since the word “recommended” is sometimes confusing given that all of these rules are of the “Recommended Item” type, the word “Optional” is a good substitute.



Keep names short and simple.

If you have a large number of rules at a high level (e.g. Product Family) you may want to try a prefix naming system to group them together, e.g. “Support - Options”, “Support - Term”, “Hardware - Advanced Config”.



Do not use phrases like “greater than”, “less than”, or “equal to”. When there is a large list of rules it is hard to tell which words are attributes and which words are operators. The exception to the above rules is “CONTAINS”. Use it and its negative ”!CONTAINS” in all caps so that you can differentiate them from the operands in the rule name.

Naming Functions

Each function should have meaningful names that accurately convey what it does. Function/method names should start with a strong action verb, e.g. convert, divide, build, etc. Get and set are often used in function names. The rest of the name should reference what the function is acting on or perhaps what it is returning. **Examples may include:**

- divideProductIntoBuckets
- buildSupportTypeQuery
- getXMLNode
- setSupportPrice

Naming Documents

There are two primary types of documents in BigMachines:

- DocEngine/Printer Friendlies

The names for Printer Friendlies should simply reflect the contents of the document. If the document is a Schedule A, the name 'Schedule A' is fitting. If it consists solely of a financial analysis then 'Financial Analysis' fits. The rule of thumb is that each

document should be easily distinguished from all of the other documents. Because of this, the more documents a site has the more important the names become.

- XSL Integration Docs

Integration Docs are a bit harder to name. Luckily QuickStart comes with well-named Documents already built in.

- **The QuickStart Template:**
  - Object - Action
- **Examples:**
  - SFDC Opportunity - Import
  - SFDC Quote - Upsert
  - SFDC Contacts - Get IDs



The Object refers to the CRM Object that is being queried or updated. The Action explains what is being done with that Object.

The most common Actions for an integration are:
-Create
-Delete
-Upsert
-Import
-Get IDs

Naming BML Variables

Often times the sheer number of variables in BML scripts causes admins to give little thought to the variable names. *Poorly chosen names can make code difficult or impossible to read.* **Types of BML variables:**

Variable Type	Description	Notes
Constant	Values never change through the entire script.	They should be ALL CAPS. Separations in words should be replaced by an underscore (i.e. LIKE_THIS) They should be included at the top of the BML script in which they are contained. They should be used for commonly-used strings and numbers. A pricing script might have the literals: • ITEM_DELIM = "|"; • FIELD_DELIM = "~"; • MAIN_LEVEL_INDEX = 3;

Variable Type	Description	Notes
		If the field delimiter changes from a tilde to another, an admin needs only to change the initialization of the variable instead of searching through the code for every single instance of a tilde. Try to use these literals when pulling values from an array. **Example:** write listPrice = pricingInfoArray[LIST_PRICE_INDEX] instead of referring to pricingInfoArray[2]
True Variable	Any variable whose value can change as the script runs.	They should be in camelCase. The first letter is lowercase, while the first letter of each element in the name is upper case, i.e. listPriceAfterStandardDiscount Avoid acronyms when possible, unless they are standard acronyms like XML, HTML, etc. When acronyms are included, they must also follow the camelCase rule (i.e. parsedXmlHtml) Make your variable names "information dense" so that the admin will have an idea of what the variable represents, without context. Keep variable names short, but not so short that they are not useful.

Naming Attributes

Follow the recommendations for variables above, when naming Configurable and Commerce Attributes. In addition, you should follow a naming convention that makes it easier to distinguish between Attributes and regular variables in a script, since they behave differently (e.g. you cannot set the value of an Attribute in a script). **Example of "`_object`" convention which is used in QuickStart:**

- **Quote:** `quoteNumber_quote`, `submittedDate_quote`
- **Line:** `listPrice_line`, `isOptional_line`
- **Model "Rack":** `server_rack`, `memory_rack`, (or simply `server_m`, `server_pl`, `server_f` etc.)

Naming Data Tables

To more easily distinguish between variables, it is recommended that you use Capitalized CamelCase when naming data tables. **Example:**

- **Table Name:** ContractPricing
- **Table Columns:** Product, Username, StartDate

RELATED TOPICS

APPLICATION BOUNDARIES AND MAGIC NUMBERS

OVERVIEW

Things You Can't Do

BML:

• Use certain special characters in a script (e.g., 0x1c), or they can corrupt the entire commerce process when the script is deployed

Configuration:

• Link directly to a specific tab in a configuration flow

User and Group Administration:

• Delete a user (instead, use Inactivate)

• Delete a group

Reporting Boundaries

One application boundary to be aware of deals with this new feature and reporting. Reporting will not convert quotes into a single currency. This means that if a Sales Rep or manager wants to do some reporting on quote value roll-ups, they will need to duplicate that report for every currency being quoted.

Magic Numbers

Hard limits on features:

- 1 numeric pricebook custom field
- 2 multi-select part custom fields
- 3 number part custom fields
- 5 custom pricebook fields
- 8 digit currency or float attributes converted to scientific notation (ex: 10,000,000 is 10e6)
- 10 saved views per user
- 10 views that a superuser can share
- 15 languages enabled at one time
- 16 digits maximum length of a Float data table column (and 16 9's is too big, 15 9's and 8 works fine).
- 18 digit maximum for the _system_transaction_counter
- 20 characters in a data table name

- 20 views a superuser can create
- 25 integer columns in a data table
- 25 float columns in a data table
- 30 pricebooks
- 30 part custom fields
- 30 values in a multi-select part custom field
- 30 conditional columns per table (Doc Engine in v11.1)
- 30 character limit for username field.
- 150 sub document data columns.
- 50 (string) columns in a data table (prior to 11.0 50 total columns)
- 80 characters displayed in a column on the Transaction Manager
- 101 main document data columns
- 150 Data tables with initial site setup (Ops can add more)
- 170 character limit for page names in Doc Engine
- 200 line items sent to SalesForce in one batch (SFDC limitation)
- 200 config array elements (50 until v10.0)
- 255 characters in a data table cell
- 255 characters in a text field(Not text area)
- 255 characters in a parts custom field - String type attribute
- 255 characters in a commerce action description
- 256 characters for the approve/reject comment in Approval Workflow
- 300 results returned from WebServices Data Table API
- 1000 config attributes referenced in a rule
- 1000 elements in a BML array
- 1000 charatcers for the Submit Comment in Approval Workflow
- 4000 characters in an attribute description

- 4000 characters in a function description
- 4000 characters in a Model/Product Line description used on the home page
- • 65000 parts exported per CSV.
- 64kb maximum length of a BML script (limit on java try block). Note that this is a limit on the length of the java code that is generated by the BML script, not the length of the BML script itself (which is typically larger).
- 999999999 maximum value of an Integer data table column
- -2,147,483,648 to 2,147,483,167 = integer variable value range

RELATED TOPICS

FO TO XLS (EXCEL)

OVERVIEW

This article shows you how to convert from XSL:FO to an Excel Spreadsheet.

ADMINISTRATION

Converting XSL:FO to Excel Spreadsheet

1. Remove Excel Gridlines from the output by adding the following to the worksheet options:

```
<x:DoNotDisplayGridLines>true</x:DoNotDisplayGridLines>
```

2. Set the default Zoom level of the spreadsheet by adding the following to the worksheet options (in this example, 75%):

```
<x:Zoom>75</x:Zoom>
```

3. Preserve Rich Text Attribute formatting on the Excel output. *The value of the Rich Text attribute stored in the Document XML will be something like:*

```
<p><em>Quo usque tandem abutere, Catalina, patienta nostra?</em><p>
<p/>
<p><strong><u>Quam diu etiam furor iste tuus nos eludet?</u></strong></p>
<p/>
<p><strong><u><em>Quem ad fidenm sese effrenata iactabit audacia?</em></u></strong></p>
```

Thankfully, the attribute's data is not enclosed in CDATA when being used for Printer Friendly/History XSL documents (however, they are for Integration XSLs). Thus, you can import the rich text attribute and treat it like a node set, and use XPath functions and expressions to traverse its contents. For example, writing the following:

```
<xsl:value-of select="$main_doc/richTextAttribute_quote/p[1]/em"/> => Quo usque tandem abutere, Catalina, patienta nostra?
```

Three templates for parsing Bold, Italic and Underlined text from a Rich Text Attribute:

```
<xsl:template match="strong" xmlns="http://www.w3.org/TR/REC-html40">
<B><xsl:apply-templates/></B>
</xsl:template>
<xsl:template match="em" xmlns="http://www.w3.org/TR/REC-html40">
<I><xsl:apply-templates/></I>
</xsl:template>
<xsl:template match="u" xmlns="http://www.w3.org/TR/REC-html40">
<U><xsl:apply-templates/></U>
</xsl:template>
```

The internal call to <xsl:apply-templates> ensures that templates are called recursively on the input to each template, until no more templates are found matching any nested nodes (i.e. plain text).

The XML namespace declaration is required within the scope of where the Excel XML tags (<B>, <U>, <I>) are being used. In this case, they are being used inside templates. If they are being hard-coded into a <Data> attribute, the XML namspace must be declared there as well. For example:

```
<Data ss:Style="aStyle" xmlns="http://www.w3.org/TR/REC-html40">
<Cell>
<B>This is bold text</B>
<B><U>This is bold and underlined text</U></B>
</Cell>
</Data>
The above templates can then be called using the <xsl:apply-templates> element:
<Data>
<Cell>
<xsl:apply-templates select="$main_doc/richTextAttribute_quote/*"/>
</Cell>
</Data>
```

RELATED TOPICS

PARSING STRINGS INTO NUMBERS

OVERVIEW

Numbers in BigMachines are often stored as strings. In order to perform any calculations with these values, the data should first be parsed into a number. Parsing data can be error prone. By being thorough and type-checking our data, we can create flexible solutions that behave reasonably when dealing with invalid inputs.

Number parsing may come into play in BML, JavaScript, XSL, or any other scripting or programming language that passes data around or accepts user input. This article will be focused solely on the mechanisms used within BML.

ADMINISTRATION

There are two BML functions that directly convert a string into a number:

```
parsedInteger = atoi("1");
parsedFloat = atof("1.1");
```

These functions provide a useful function, but will raise errors when used directly without type-checking first. *For instance, consider the following inputs:*

```
parsedInteger = atoi("");
parsedFloat = atof("1.1%");
```

Passing in these non-numeric strings will cause the functions to throw "invalid literal" errors.

To deal with this, the isnumber function is available. This function takes a string, and returns true if the string can be converted into a float. This function can help you identify invalid data; it is up to you to determine the best way to handle invalid inputs. *Out of the box, isnumber is all that you need to safely use the atof function:*

```
value = "1.0";
new_value = 0;
if(isnumber(value)) {
// conversion will only occur if the value is safe
new_value = atof(value);
}
```

isnumber() is not sufficient when using atoi. **Consider the following:**

```
value = "1.0";
new_value = 0;
if(isnumber(value)) {
// "1.1" will pass the isnumber check, but is an invalid input for atoi
new_value = atoi(value);
}
```

Because of this drawback, **best practice is to never use atoi**. Instead, use the safe atof function, in conjunction with isnumber. To get an integer, you can then caste the float value using the "integer" function:

```
value = "1.0";
new_value = 0;
if(isnumber(value)) {
//safely caste to integer, after translating to float
new_value = integer(atof(value));
}
```

We recommend using util libraries to wrap up these tests for you; it makes for cleaner code and is easier to maintain.

customAtoi Util Library

```
/*
BML util (customAtoi)
- To convert the string into integer.
- if the input is not a valid number, the application will take the default value input and return back as a result.
param:
inputValue - String
defaultValue - integer
result:
Integer
*/
result = defaultValue;
if ( isnumber(inputValue) ) {
result = integer(atof(inputValue));
}
return result;
```

Usage:

```
//returns integer 1
value = util.customAtof("1.1", "Invalid Value");
//returns error value "Invalid Value"
value = util.customAtof("1%", "Invalid Value");
```

customAtof Util Library

```
/*
BML util (customAtof)
- To convert the string into integer.
- if the input is not a valid number, the application will take the default value input and return back as a result.
```

```
param:
inputValue - String
defaultValue - integer
result:
Float
*/
result = defaultValue;
if ( isnumber(inputValue) ) {
result = atof(inputValue);
}
return result;
```

Usage:

```
//returns float 1.1
value = util.customAtof("1.1", "Invalid Value");
//returns error value "Invalid Value"
value = util.customAtof("1.1%", "Invalid Value");
```

TROUBLESHOOTING

Using these safe methods of parsing numbers will keep errors from occurring in the face of invalid data. But this doesn't mean that your problems are over. There is a reason that the data was invalid to begin with, and it is up to you to determine if that is a serious problem, and what to do about it.

Sometimes it may be appropriate to "throw" an error, so that the admin or other people maintaining the site will have visibility to the problem. An effective way to do this is to write into a validation attribute. This is simple to accomplish - in your advanced modification function, you write into a text attribute. In the advance validation function, you check that function and show an error if there is any value in it.

In the advanced modify:

```
...
int_value = customAtoi(value, "There was an invalid input for value: " + value);
if(!isnumber(int_value) {
resultString = resultString + "1~validationMessage_quote~" + int_value;
}
...
In the advanced validation:
...
resultString = resultString + validationMessage_quote;
...
```

RELATED TOPICS

STORING DATA IN DELIMITED STRINGS

OVERVIEW

What

A delimited string is a way of representing data using specified sequences of characters to separate individual entries. Delimited strings can be an attractive solution for storing a dynamic set of data. Using delimited strings, you might serialize a dictionary or array, so that you can store or share the information. **Example of a simple delimited string:**

```
!!KEY!!value!!KEY2!!value2
```

Why

Delimited strings are useful for data that doesn't fit into configurable, line item, or main doc attributes. Please be sure to completely eliminate those options before going with a delimited string, since it is much easier to maintain and troubleshoot data that is stored in default BigMachines data structures. Delimited strings can also be used when sending information to external services such as a CGI or Web Service.

Where

When dealing with delimited strings, the data will often be parsed and manipulated within BML, preferably within util functions to limit code duplication. The strings may also be both read and written from within XSL scripts, for both Integration and RTF/PDF Output. The strings may be sent to external systems through URL parameters or through other WebServices, including WebMethods.

ADMINISTRATION

Before you begin

1. Delimited Strings can Run out of Space. Make sure your attribute lengths are long enough, and watch out for data tables and their 255 character limit.
2. Bad Delimiters Corrupt Data. Characters like tildes, pipes, angle brackets, and ampersands can clash with XML, HTML, URL parameters, and other delimited strings.
3. Delimited Strings can be Difficult to Maintain. Do what you can to make your string descriptive to human troubleshooters.

Choose your delimiters

Make your delimiters descriptive, and consider your source data in picking delimiters that won't cause conflicts. Use delimiters that don't look like your data. Combining words into your delimiters can be very effective at conveying information, while avoiding collisions. **Examples:**

- !ROW!!VAL!15!DISC!0.5!ROW!!VAL!15!DISC!0.2
-)(_value_15_discount_0.5)(_value_15_discount_0.2



Your choice of delimiter should consider what, if any, external systems or protocols might touch your data. For instance, ampersands (&) can cause problems if sent through a URL. Angle brackets (>,<) can cause problems when dropped into an HTML or XML string.

Safe Basic Separators: ,**::_some_keyword_



Notice that the above string is easy to parse using string manipulation, and also descriptive to human readers.

Put delimiters into an attribute

In order to make it simple to manage and maintain your delmiters, it is best to store them in BigMachines attributes. Then if you need to make chagnes, it is simply a matter of updating the attribute, and all of your code will then be in compliance.

Create the string

Once you know the format for the delimited strings, put the string into the format as early in the process as possible, in order to avoid inconsistency when dealing with the given values. If the data is coming in from a SalesForce integration, for instance, then consider building the string directly in the XSL Parser, before it is even stored in BigMachines. Use a util library to create the delimited string, and use a key/value pairing system as often as possible.

It's good practice to break the delimiter out into its own variable, to make it easier to work with. **For instance:**

```
delim=columnDelimiter_quote;
// create a delimited string, using key/value pairs
delim_string = "totalDiscount" + delim + totalDiscount_quote
+ delim + "discountReason" + delim + discountReason_quote;
```

Retrieve values from the string

Use a util library to retrieve the string as often as possible. You will basically be using the BML “split” function to process the string. Don't be too trusting. The data coming in might look different than you expect. Write your code defensively, to deal with common problems.

```
ret = "Error parsing string in util library: " + delimited_string;
delim=delimiter_quote;
delimited_string = "discount_value"+delim+"10.0"+delim+"discount_percent"+delim+"0.5";
parsed_array = split(delim, delimited_string);
//test for length first!
if(sizeofarray(parsed_array) > 10) {
```

```
discount_value = parsed_array[1];
discount_percent = parsed_array[3];
// validate numericality of data before atoi/atof
if(isnumber(discount_value) and isnumber(discount_percent)) {
full_value = atoi(discount_value) / atoi(discount_percent);
ret = full_value;
}
}
return ret;
```

NOTES



Test your data thoroughly before parsing.

Always be thorough when performing number parsing, watching out for Number Parsing Errors.

Watch out for Null Pointer Exceptions.



When retrieving values from a string,don't assume the values are formed perfectly.

RELATED TOPICS

PARTICIPANT PROFILE BEST PRACTICES

OVERVIEW

Participant Profiles are created for each step in the workflow. *For each profile, you can define:*

- **Document Views:** define whether actions and groups are shown or hidden, and whether attributes are read-only, editable, or hidden
- **Transition Rules:** define what workflow step will be next and set up email notifications
- **User Access Rights:** control who will be assigned to this profile once a transaction reaches this step (you can also control user access to a profile using the Advanced Forwarding Rules for the Step)

Assigning Users to Participant Profiles

There are 2 ways to assign users to Participant Profiles:

- User Access Rights

  Here you can assign Participant Profiles based on:

  - Company Type and User Type
  - Groups

  Example: you could assign all `FullAccessWithESales` users to a profile called Administrator, and all members of the Manager group access to the Approver profile for all quotes.

  The Participant Profile seen by the user is the FIRST profile which evaluates to True. For this reason, you usually want to make an Admin Profile first.

- Advanced Forwarding Rules

  You can use the Advanced Forward Rules to assign Participant Profiles to Groups, based on attributes in the quote. Participant Profiles will only be assigned when the step changes.

  Example: if approvers are assigned to quotes according to a user hierarchy, you can define a quote attribute to store the name of the approver group (e.g., Manager-East, Manager-West). In the Forward Rule, you can use that attribute to determine which groups to assign to the Approver profile for that quote.

  The Expected format of return value is: group~company~participantProfile [

|group~company~participantProfile]*\

Note that Participant Profiles are assigned according to Groups and User Types. *You can't assign profiles to individual users by username.*

Defining Document Views

Document views determine what is visible to users in the participant profile. Choices for the document views depend on the entity. For tabs, the options are Show or Hide. For attributes, the choices are Read/Write, Read-Only, and Hide.

You can also adjust these setting from the attribute or action, enabling you to quickly adjust the behavior of the same entity across all steps.

If a tab is hidden, all attributes and actions within the tab are hidden.

Defining Transitions and Notifications

Notifications: Notification emails can be defined for each transition; each recipient must receive the same message. Recipients can be defined using the Participant Profiles for the current step and next step, or by defining an advanced function to specify the email address.

Transitions: Transitions rules are defined per action per profile. Several transitions may be defined for the same profile and action. The first transition to evaluate is true will take effect.

Transition rules can have the same step as a destination. Use this to send notifications without changing steps (e.g. a reminder email).

Since notification emails must be defined for each transition, the same message may be defined in many places. By using XML templates and variables in the subject line as much as possible, you can



reduce the number of places you will need to update to make changes to the messages



XML notifications are sent to the partner server specified in Admin/Integration Settings. This allows integration with ERP and other non-CRM systems at specified transitions.



Always put an always true rule at the bottom of a set to avoid errors in logs.

NOTES



Check performer steps, forwarding rules, and rights based on user type.



Profiles are evaluated in the order in which they appear.

When a user forwards a transaction manually from the transaction manager, the recipient will get the same access rights as the sender.

RELATED TOPICS

PARTICIPANT PROFILE BEST PRACTICES

OVERVIEW

Participant Profiles are created for each step in the workflow. *For each profile, you can define:*

- **Document Views:** define whether actions and groups are shown or hidden, and whether attributes are read-only, editable, or hidden
- **Transition Rules:** define what workflow step will be next and set up email notifications
- **User Access Rights:** control who will be assigned to this profile once a transaction reaches this step (you can also control user access to a profile using the Advanced Forwarding Rules for the Step)

Assigning Users to Participant Profiles

There are 2 ways to assign users to Participant Profiles:

- User Access Rights

  Here you can assign Participant Profiles based on:

  - Company Type and User Type
  - Groups

  Example: you could assign all `FullAccessWithESales` users to a profile called Administrator, and all members of the Manager group access to the Approver profile for all quotes.

  The Participant Profile seen by the user is the FIRST profile which evaluates to True. For this reason, you usually want to make an Admin Profile first.

- Advanced Forwarding Rules

  You can use the Advanced Forward Rules to assign Participant Profiles to Groups, based on attributes in the quote. Participant Profiles will only be assigned when the step changes.

  Example: if approvers are assigned to quotes according to a user hierarchy, you can define a quote attribute to store the name of the approver group (e.g., Manager-East, Manager-West). In the Forward Rule, you can use that attribute to determine which groups to assign to the Approver profile for that quote.

  The Expected format of return value is: group~company~participantProfile [

|group~company~participantProfile]*\

Note that Participant Profiles are assigned according to Groups and User Types. *You can't assign profiles to individual users by username.*

Defining Document Views

Document views determine what is visible to users in the participant profile. Choices for the document views depend on the entity. For tabs, the options are Show or Hide. For attributes, the choices are Read/Write, Read-Only, and Hide.

You can also adjust these setting from the attribute or action, enabling you to quickly adjust the behavior of the same entity across all steps.

If a tab is hidden, all attributes and actions within the tab are hidden.

Defining Transitions and Notifications

Notifications: Notification emails can be defined for each transition; each recipient must receive the same message. Recipients can be defined using the Participant Profiles for the current step and next step, or by defining an advanced function to specify the email address.

Transitions: Transitions rules are defined per action per profile. Several transitions may be defined for the same profile and action. The first transition to evaluate is true will take effect.

Transition rules can have the same step as a destination. Use this to send notifications without changing steps (e.g. a reminder email).

Since notification emails must be defined for each transition, the same message may be defined in many places. By using XML templates and variables in the subject line as much as possible, you can



reduce the number of places you will need to update to make changes to the messages



XML notifications are sent to the partner server specified in Admin/Integration Settings. This allows integration with ERP and other non-CRM systems at specified transitions.



Always put an always true rule at the bottom of a set to avoid errors in logs.

NOTES



Check performer steps, forwarding rules, and rights based on user type.



Profiles are evaluated in the order in which they appear.

When a user forwards a transaction manually from the transaction manager, the recipient will get the same access rights as the sender.

RELATED TOPICS

USING JAVASCRIPT IN BIGMACHINES

OVERVIEW

What

JavaScript is a client-side interpreted language. It is available in most browsers, and can be used to augment a site's functionality.

Why

The decision to do custom development on a site is one that should never be made lightly. The risks non-standard work introduce to the long-term success of a project are significant, and the implications and downsides of the decision can be tough to predict or plan for. Before considering JavaScript as an option, make sure to perform a proper risk assessment. Include all the stakeholders of the project in the discussion, be aware of the risks and open a discussion to pursue alternative options. **Considerations:**

1. Custom Development is a form of tribal knowledge. *It creates the need for a specialist in an otherwise standardized system.*
2. Upgrades to the BigMachines application may conflict with your add-on work.
3. Bugs in custom development work may have serious implications, including data corruption and loss.
4. Custom work may interfere with "Standard" BigMachines functionality, including admin tasks and troubleshooting.

Where

JavaScript should be written within the JavaScript Framework. *The Framework has a few primary objectives:*

1. Centralize all JavaScript in the File Manager.
2. Avoid the creation of global variables.
3. Load code only on the page that it will be run.
4. Minimize the impact of bugs in JavaScript on the rest of the site.
5. Minimize the effort involved in disabling or removing the code in the future.

How

Before you begin: LEARN JAVASCRIPT! Though it shares many characteristics of other C-style languages, JavaScript has many language features that can be surprising, and can cause serious issues if misused. *Some great resources:*

1. Eloquent JavaScript is a good JavaScript tutorial for beginners.
2. HTML Dog is a good resource for web-development best practices.
3. Mozilla Developer Network is a great CSS, HTML, and JavaScript reference.

ADMINISTRATION

Build for failure

When approaching a JavaScript solution to a problem, it is important to consider that there are many different variables that could cause your features to break. In fact, between Browser compatibility issues and upgrades to BigMachines, it is unreasonable to treat anything in the HTML of a page as constant or immutable. To deal with this uncertainty, design code that fails gracefully. This means different things in different situations. *Consider the following code:*

```
function getPrice() {
var value = document.getElementById("netPrice_quote").value;
return value;
}
```

If this code is run on a page that doesn't have `"netPrice_quote"` available, then it will throw a `TypeError`, because `getElementById` will return null. This means that changes to the layout of the page will cause the entire BigMachines application to break. *You can deal with this by using jQuery to select the element. If `netPrice_quote` isn't on the layout, then this code will not throw an error:*

```
function getPrice() {
var value = jQuery("#netPrice_quote").val();
return value;
}
```

But just because the code doesn't throw an error, doesn't necessarily mean it is failing gracefully. It's possible that not having the net price will create serious problems for the application, such as data loss or corrupted quotes. *If this is the case, then a loud, complete failure may be appropriate:*

```
function getPrice() {
var value = jQuery("#netPrice_quote").val();
return value;
}
function usePriceForCalculation() {
var price = getPrice();
//if net price is not available, don't allow this script to run and warn the user
if(price === undefined) {
jQuery("body").prepend("<h2>There is an error on this quote. Please contact your system administrator.</h2>");
throw new Error("Custom work per Case #00012345. Net Price was not available for a critical calculation.");
}
// ...otherwise use it
}
```

Don't use inline JavaScript

JavaScript is largely an event-driven language. You can bind behavior to a web page in the form of loading events, click events, change events, and more. Many resources will recommend that you use "inline" JavaScript to accomplish this. **For example:**

```
<body onload="myJavascriptFunction()">
<input id="my-button" type="button" onclick="myJavascriptFunction()"/>
<select id="my-select" onchange="myJavascriptFunction()"/>
<a id="my-link" href="javascript:myJavascriptFunction()">
```

This method of event binding violates industry best practice, and creates code that is difficult to maintain and modify. The preferred way to accomplish this task is to select the elements you want to fire an event on, and then bind the event from within your code. *Using jQuery, the above examples would be accomplished like this:*

```
jQuery(document).ready(myJavascriptFunction)
jQuery("#my-button").click(myJavascriptFunction);
jQuery("#my-select").change(myJavascriptFunction);
jQuery("#my-link").click(myJavascriptFunction);
```

This approach has the clear benefit of separating your logic from your data, which makes it easier to maintain, and also allows you to centralize all of your code into a single file. It also means that the function being called doesn't have to be globally available.



It is intentional that code cannot be called from `<a onclick="...">`

Scope all variables

When you use or define a variable in JavaScript, it is very important not to create new globals. Global variables introduce maintenance issues and unpredictability to your code, and can also conflict with other JavaScript that is already existing on the page. These conflicts are difficult to detect and can have subtle and surprising results. In order to avoid creating globals, there are a few things you have to be aware of:

Use the var keyword when declaring new variables

Variables declared without the var keyword will not be scoped, no matter where they are in the program. This is bad, because you risk overwriting or colliding with other global variables that already exist on the page. And if a variable of the same name does not exist, then you will have inadvertently exposed your variable to the global namespace.

```
function createsGlobals() {
var non_global = "This variable will only be available within the createsGlobals function";
is_global = "This variable will be global, or will overwrite existing variables of the same name";
}
```

Functions are the only blocks with scope

In many other C-like languages, anything within { and } blocks are scoped. This is not the case in JavaScript.

```
if(true) {
var is_global = "This variable will be available outside the if block, even though I used 'var'";
// this function will be available from outside the if block
function global_function() {
```

```
var safe = "This variable is private to global_function. Functions are the only blocks with scope.";
}
//variables i and ii will be globally available
for(var i = 0, ii = 10; i<ii; i++) {
var is_global = "This variable will overwrite the previous is_global, even though I used 'var'";
}
}
```

---

jQuery.ready, require, and anonymous functions

The JavaScript Framework uses RequireJS to manage dependencies and scope. In terms of scope management, it is similar to putting code into jQuery's "ready" function, in that as long as you use the var keyword properly none of your code will be available globally:

```
// none of the variables in this example will collide, even though they have identical names
jQuery(document).ready(function() {
var safe = "scoped variable";
var scoped_function = function() { }
});
require([], function() {
var safe = "a different scoped variable";
var scoped_function = function() { }
});
```

The same effect can be accomplished with an anonymous function. Just put () after the closing } bracket to fire the function immediately:

```
(function() {
var inner = "scoped variable";
require([], function() {
var inner = "a different scoped variable";
});
}());
```

This example creates no new global variables.

Declare your variables upfront

A good way to deal with confusing scope within JavaScript is to declare all of your variables at the top of each function, using a single comma-separated "var" statement:

```
function sum_array(arg) {
var result, i, ii;
for(i=0, ii=arg.length; i<ii; i++) {
result += arg[i];
```

```
}
return result;
};
```

Treat native objects as private

You will occasionally find suggested solutions where a JavaScript native object is extended with a new method. *You might see something like:*

```
// DANGEROUS! Don't do this:
//adding a method to the "String" object...
String.prototype.trim = String.prototype.trim || function() { }
//extending the "Array" object...
Array.prototype.remove = Array.prototype.remove || function() { }
```

Within the BigMachines application, this is inappropriate. This creates fragile and difficult to maintain code, and there is a risk of a conflict with other libraries and code. *If you want to use JavaScript functions that exist in some browsers and not in others, write a "wrapper" function rather than extending the object:*

```
(function() {
function is_array(obj) {
// Use the native function if it exists
if(typeof Array.isArray === "function") {
return Array.isArray(obj);
}
// my own custom code to test for array goes here
// ...
}
if(is_array(new Array())) {
//do stuff...
}
}());
```

Treat BigMachines JavaScript as private

The BigMachines Development group has written a lot of JavaScript that is available publicly in the global namespace. These functions should be treated as a private interface - do not call them, and do not modify them. Development may choose at anytime to refactor, rewrite, or otherwise improve their functionality. If they do so, then any code that you have written to rely on those functions will suddenly stop working, and it may be very difficult to recreate the functionality.

Use brackets

JavaScript lets you skip the brackets for one line blocks:

```
if(true) return;
for(var i=0, ii=arr.length; i<ii; i++)
```

```
count+=1;
```

Don't do this! This is a maintenance hazard, and makes it very easy for the next programmer to come along and do this:

```
if(true) return;
for(var i=0, ii=arr.length; i<ii; i++)
count+=1;
// oops! we are no longer in the loop!
alert(arr[i]);
```

Use jQuery for DOM selection and manipulation

jQuery provides robust, dependable methods for binding events, selecting HTML elements, and modifying the page. When it is available, use it. For more information, see Using jQuery in BigMachines.

Use functions and objects

When using JavaScript's flexible functions and objects, your code can become descriptive and maintainable. Consider the example of a script that populates billing information in a commerce quote, based on some data from a delimited string. It may look something like this:

```
//this example uses require.
//see the JavaScript Framework for more information.
require([], function() {
require.ready(function() {
var delimited_string, delimiter, ship_array, ship_object = {}, $shipping_form, i, ii, temp;
delimited_string = jQuery("#1_billingDelimitedString_quote").val();
delimiter = jQuery("#1_billingDelimiter_quote").val();
//bail out if we can't find our data
if(!delimited_string || !delimiter) { return; }
ship_data = delimited_string.split(delimiter);
for(i=0, ii=ship_data.length; i<ii; i++) {
//create a dictionary of key/values
}
//populate the fields accordingly
$shipping_form = jQuery("form[name='shipping']");
jQuery("input[name='name']", $shipping_form).val(ship_object["name"]);
//... and so on
});
});
```

The problem with this code is that one function has three distinct jobs: get the data, parse the data, and present the data. By splitting this up the code becomes much more flexible:

```
require([], function() {
```

```
var addresses = {
parse_data: function(str, delimiter) {
// return the parsed data
},
get_parsed_data: function(data_selector, delim_selector) {
//gets the strings, and pass them to parse_data
},
fill_form: function($form, object) {
// populate the form
}
string_to_form: function(data_selector, delim_selector, form_selector) {
var data, $shipping_form;
$shipping_form = jQuery(form_selector);
data = addresses.get_parsed_data(data_selector, delim_selector);
addresses.fill_form($shipping_form, data);
}
};
require.ready(function() {
addresses.string_to_form("#1_shippingDelimitedString", "#1_shippingDelimiter", "form[name='shipping']");
});
});
```

Now within require.ready, we call `string_to_form`, which does a pretty good job of describing the goal of the app. And by setting parameters in the functions, we can point at different data sources and forms:

```
...
require.ready(function() {
// use the same code to populate both shipping and billing info
addresses.string_to_form("#1_shippingDelimitedString", "#1_shippingDelimiter", "form[name='shipping']");
addresses.string_to_form("#1_billingDelimitedString", "#1_billingDelimiter", "form[name='billing']");
});
...
```

Use === and !==

JavaScript has two types of comparison operators: double and triple. Briefly, the difference is that the double operator attempts to cast the type of the compared values, while the triple operator does not. This can lead to some surprises. *For instance:*

```
false == '0'; //true
'' == '0'; //false
0 == '0'; //true
0 == ''; //true
null == false; //false
```

```
null == undefined; //true
```

The rules behind this type-casting are unusual, and not very memorable. *Use the === and !== operators to remove ambiguity:*

```
false === '0';// false
'' === '0'; //false
0 === '0'; //false
0 === ''; //false
null === false; //false
null === undefined; //false
```

Test your code

Your JavaScript will be running in a lot of different environments, and it's important to test it in every environment that your users will be running. The Framework provides QUnit as a way to test your code.

TROUBLESHOOTING

Errors & exceptions

It can be tempting to use try/catch blocks as a way to shield the user from errors. This can make it very difficult to locate the source of a problem some time in the future.

```
(function() {
try {
my_array[3].obj.value = "Uh-oh, this is dangerous.";
} catch(e) {/* Do nothing - just hide it */}
}());
```

The problem with try/catch is that it is often far too broad; it can cover up errors that you didn't anticipate would occur. Instead of throwing away exceptions, take steps to avoid specific errors. If other issues come up, then it will be much easier to single them out and deal with them.

```
(function() {
if(my_array.length < 4) { return; }
if(!my_array[3].obj) { return; }
my_array[3].obj.value = "This is more specific, and easier to troubleshoot.";
}());
```

I can't call my code from the console!

Code within the Framework is non-global, which means that directly calling the function from the console is not an option. *To expose the code, you have two options.* You can call "require" from the console:

```
//directly in the console...
require(["my_module"], function(module) {
//test module directly:
module.some_function();
```

```
});
```

Or you can "export" your code to the global namespace from within the File Manager. Be sure to add a flag as this is only appropriate for testing purposes:

```
//within the file manager...
require([], function() {
var addresses = {}, debug = true;
addresses.parse_data = function() {...}
...
if(debug === true) {
window.addresses = addresses;
}
});
```

The above example will make "addresses" globally available when the debug variable is set to true. This will allow you to do this directly in the console:

```
addresses.parse_data();
```

NOTES



For more information regarding BigMachines and the JavaScript Framework, see GS.COE.JA.18 - JavaScript Framework 2.0 Upgrade Kit.



In Firefox, ensure that you have Firebug available. In Chrome and IE, the developer tools are useful for debugging JavaScript.

You will want to use the "Script" tab in these tools, and then set break points within your code. When you reload the page, you will be able to step through the code at the point you chose.

RELATED TOPICS

USING JAVASCRIPT IN BIGMACHINES

OVERVIEW

What

JavaScript is a client-side interpreted language. It is available in most browsers, and can be used to augment a site's functionality.

Why

The decision to do custom development on a site is one that should never be made lightly. The risks non-standard work introduce to the long-term success of a project are significant, and the implications and downsides of the decision can be tough to predict or plan for. Before considering JavaScript as an option, make sure to perform a proper risk assessment. Include all the stakeholders of the project in the discussion, be aware of the risks and open a discussion to pursue alternative options. **Considerations:**

1. Custom Development is a form of tribal knowledge. *It creates the need for a specialist in an otherwise standardized system.*
2. Upgrades to the BigMachines application may conflict with your add-on work.
3. Bugs in custom development work may have serious implications, including data corruption and loss.
4. Custom work may interfere with "Standard" BigMachines functionality, including admin tasks and troubleshooting.

Where

JavaScript should be written within the JavaScript Framework. *The Framework has a few primary objectives:*

1. Centralize all JavaScript in the File Manager.
2. Avoid the creation of global variables.
3. Load code only on the page that it will be run.
4. Minimize the impact of bugs in JavaScript on the rest of the site.
5. Minimize the effort involved in disabling or removing the code in the future.

How

Before you begin: LEARN JAVASCRIPT! Though it shares many characteristics of other C-style languages, JavaScript has many language features that can be surprising, and can cause serious issues if misused. *Some great resources:*

1. Eloquent JavaScript is a good JavaScript tutorial for beginners.
2. HTML Dog is a good resource for web-development best practices.
3. Mozilla Developer Network is a great CSS, HTML, and JavaScript reference.

ADMINISTRATION

Build for failure

When approaching a JavaScript solution to a problem, it is important to consider that there are many different variables that could cause your features to break. In fact, between Browser compatibility issues and upgrades to BigMachines, it is unreasonable to treat anything in the HTML of a page as constant or immutable. To deal with this uncertainty, design code that fails gracefully. This means different things in different situations. *Consider the following code:*

```
function getPrice() {
var value = document.getElementById("netPrice_quote").value;
return value;
}
```

If this code is run on a page that doesn't have `"netPrice_quote"` available, then it will throw a `TypeError`, because `getElementById` will return null. This means that changes to the layout of the page will cause the entire BigMachines application to break. *You can deal with this by using jQuery to select the element. If `netPrice_quote` isn't on the layout, then this code will not throw an error:*

```
function getPrice() {
var value = jQuery("#netPrice_quote").val();
return value;
}
```

But just because the code doesn't throw an error, doesn't necessarily mean it is failing gracefully. It's possible that not having the net price will create serious problems for the application, such as data loss or corrupted quotes. *If this is the case, then a loud, complete failure may be appropriate:*

```
function getPrice() {
var value = jQuery("#netPrice_quote").val();
return value;
}
function usePriceForCalculation() {
var price = getPrice();
//if net price is not available, don't allow this script to run and warn the user
if(price === undefined) {
jQuery("body").prepend("<h2>There is an error on this quote. Please contact your system administrator.</h2>");
throw new Error("Custom work per Case #00012345. Net Price was not available for a critical calculation.");
}
// ...otherwise use it
}
```

Don't use inline JavaScript

JavaScript is largely an event-driven language. You can bind behavior to a web page in the form of loading events, click events, change events, and more. Many resources will recommend that you use "inline" JavaScript to accomplish this. **For example:**

```
<body onload="myJavascriptFunction()">
<input id="my-button" type="button" onclick="myJavascriptFunction()"/>
<select id="my-select" onchange="myJavascriptFunction()"/>
<a id="my-link" href="javascript:myJavascriptFunction()">
```

This method of event binding violates industry best practice, and creates code that is difficult to maintain and modify. The preferred way to accomplish this task is to select the elements you want to fire an event on, and then bind the event from within your code. *Using jQuery, the above examples would be accomplished like this:*

```
jQuery(document).ready(myJavascriptFunction)
jQuery("#my-button").click(myJavascriptFunction);
jQuery("#my-select").change(myJavascriptFunction);
jQuery("#my-link").click(myJavascriptFunction);
```

This approach has the clear benefit of separating your logic from your data, which makes it easier to maintain, and also allows you to centralize all of your code into a single file. It also means that the function being called doesn't have to be globally available.



It is intentional that code cannot be called from `<a onclick="...">`

Scope all variables

When you use or define a variable in JavaScript, it is very important not to create new globals. Global variables introduce maintenance issues and unpredictability to your code, and can also conflict with other JavaScript that is already existing on the page. These conflicts are difficult to detect and can have subtle and surprising results. In order to avoid creating globals, there are a few things you have to be aware of:

Use the var keyword when declaring new variables

Variables declared without the var keyword will not be scoped, no matter where they are in the program. This is bad, because you risk overwriting or colliding with other global variables that already exist on the page. And if a variable of the same name does not exist, then you will have inadvertently exposed your variable to the global namespace.

```
function createsGlobals() {
var non_global = "This variable will only be available within the createsGlobals function";
is_global = "This variable will be global, or will overwrite existing variables of the same name";
}
```

Functions are the only blocks with scope

In many other C-like languages, anything within { and } blocks are scoped. This is not the case in JavaScript.

```
if(true) {
var is_global = "This variable will be available outside the if block, even though I used 'var'";
// this function will be available from outside the if block
function global_function() {
```

```
var safe = "This variable is private to global_function. Functions are the only blocks with scope.";
}
//variables i and ii will be globally available
for(var i = 0, ii = 10; i<ii; i++) {
var is_global = "This variable will overwrite the previous is_global, even though I used 'var'";
}
}
```

---

jQuery.ready, require, and anonymous functions

The JavaScript Framework uses RequireJS to manage dependencies and scope. In terms of scope management, it is similar to putting code into jQuery's "ready" function, in that as long as you use the var keyword properly none of your code will be available globally:

```
// none of the variables in this example will collide, even though they have identical names
jQuery(document).ready(function() {
var safe = "scoped variable";
var scoped_function = function() { }
});
require([], function() {
var safe = "a different scoped variable";
var scoped_function = function() { }
});
```

The same effect can be accomplished with an anonymous function. Just put () after the closing } bracket to fire the function immediately:

```
(function() {
var inner = "scoped variable";
require([], function() {
var inner = "a different scoped variable";
});
}());
```

This example creates no new global variables.

Declare your variables upfront

A good way to deal with confusing scope within JavaScript is to declare all of your variables at the top of each function, using a single comma-separated "var" statement:

```
function sum_array(arg) {
var result, i, ii;
for(i=0, ii=arg.length; i<ii; i++) {
result += arg[i];
```

```
}
return result;
};
```

Treat native objects as private

You will occasionally find suggested solutions where a JavaScript native object is extended with a new method. *You might see something like:*

```
// DANGEROUS! Don't do this:
//adding a method to the "String" object...
String.prototype.trim = String.prototype.trim || function() { }
//extending the "Array" object...
Array.prototype.remove = Array.prototype.remove || function() { }
```

Within the BigMachines application, this is inappropriate. This creates fragile and difficult to maintain code, and there is a risk of a conflict with other libraries and code. *If you want to use JavaScript functions that exist in some browsers and not in others, write a "wrapper" function rather than extending the object:*

```
(function() {
function is_array(obj) {
// Use the native function if it exists
if(typeof Array.isArray === "function") {
return Array.isArray(obj);
}
// my own custom code to test for array goes here
// ...
}
if(is_array(new Array())) {
//do stuff...
}
}());
```

Treat BigMachines JavaScript as private

The BigMachines Development group has written a lot of JavaScript that is available publicly in the global namespace. These functions should be treated as a private interface - do not call them, and do not modify them. Development may choose at anytime to refactor, rewrite, or otherwise improve their functionality. If they do so, then any code that you have written to rely on those functions will suddenly stop working, and it may be very difficult to recreate the functionality.

Use brackets

JavaScript lets you skip the brackets for one line blocks:

```
if(true) return;
for(var i=0, ii=arr.length; i<ii; i++)
```

```
count+=1;
```

Don't do this! This is a maintenance hazard, and makes it very easy for the next programmer to come along and do this:

```
if(true) return;
for(var i=0, ii=arr.length; i<ii; i++)
count+=1;
// oops! we are no longer in the loop!
alert(arr[i]);
```

Use jQuery for DOM selection and manipulation

jQuery provides robust, dependable methods for binding events, selecting HTML elements, and modifying the page. When it is available, use it. For more information, see Using jQuery in BigMachines.

Use functions and objects

When using JavaScript's flexible functions and objects, your code can become descriptive and maintainable. Consider the example of a script that populates billing information in a commerce quote, based on some data from a delimited string. It may look something like this:

```
//this example uses require.
//see the JavaScript Framework for more information.
require([], function() {
require.ready(function() {
var delimited_string, delimiter, ship_array, ship_object = {}, $shipping_form, i, ii, temp;
delimited_string = jQuery("#1_billingDelimitedString_quote").val();
delimiter = jQuery("#1_billingDelimiter_quote").val();
//bail out if we can't find our data
if(!delimited_string || !delimiter) { return; }
ship_data = delimited_string.split(delimiter);
for(i=0, ii=ship_data.length; i<ii; i++) {
//create a dictionary of key/values
}
//populate the fields accordingly
$shipping_form = jQuery("form[name='shipping']");
jQuery("input[name='name']", $shipping_form).val(ship_object["name"]);
//... and so on
});
});
```

The problem with this code is that one function has three distinct jobs: get the data, parse the data, and present the data. By splitting this up the code becomes much more flexible:

```
require([], function() {
```

```
var addresses = {
parse_data: function(str, delimiter) {
// return the parsed data
},
get_parsed_data: function(data_selector, delim_selector) {
//gets the strings, and pass them to parse_data
},
fill_form: function($form, object) {
// populate the form
}
string_to_form: function(data_selector, delim_selector, form_selector) {
var data, $shipping_form;
$shipping_form = jQuery(form_selector);
data = addresses.get_parsed_data(data_selector, delim_selector);
addresses.fill_form($shipping_form, data);
}
};
require.ready(function() {
addresses.string_to_form("#1_shippingDelimitedString", "#1_shippingDelimiter", "form[name='shipping']");
});
});
```

Now within require.ready, we call `string_to_form`, which does a pretty good job of describing the goal of the app. And by setting parameters in the functions, we can point at different data sources and forms:

```
...
require.ready(function() {
// use the same code to populate both shipping and billing info
addresses.string_to_form("#1_shippingDelimitedString", "#1_shippingDelimiter", "form[name='shipping']");
addresses.string_to_form("#1_billingDelimitedString", "#1_billingDelimiter", "form[name='billing']");
});
...
```

Use === and !==

JavaScript has two types of comparison operators: double and triple. Briefly, the difference is that the double operator attempts to cast the type of the compared values, while the triple operator does not. This can lead to some surprises. *For instance:*

```
false == '0'; //true
'' == '0'; //false
0 == '0'; //true
0 == ''; //true
null == false; //false
```

```
null == undefined; //true
```

The rules behind this type-casting are unusual, and not very memorable. *Use the === and !== operators to remove ambiguity:*

```
false === '0';// false
'' === '0'; //false
0 === '0'; //false
0 === ''; //false
null === false; //false
null === undefined; //false
```

Test your code

Your JavaScript will be running in a lot of different environments, and it's important to test it in every environment that your users will be running. The Framework provides QUnit as a way to test your code.

TROUBLESHOOTING

Errors & exceptions

It can be tempting to use try/catch blocks as a way to shield the user from errors. This can make it very difficult to locate the source of a problem some time in the future.

```
(function() {
try {
my_array[3].obj.value = "Uh-oh, this is dangerous.";
} catch(e) {/* Do nothing - just hide it */}
}());
```

The problem with try/catch is that it is often far too broad; it can cover up errors that you didn't anticipate would occur. Instead of throwing away exceptions, take steps to avoid specific errors. If other issues come up, then it will be much easier to single them out and deal with them.

```
(function() {
if(my_array.length < 4) { return; }
if(!my_array[3].obj) { return; }
my_array[3].obj.value = "This is more specific, and easier to troubleshoot.";
}());
```

I can't call my code from the console!

Code within the Framework is non-global, which means that directly calling the function from the console is not an option. *To expose the code, you have two options.* You can call "require" from the console:

```
//directly in the console...
require(["my_module"], function(module) {
//test module directly:
module.some_function();
```

```
});
```

Or you can "export" your code to the global namespace from within the File Manager. Be sure to add a flag as this is only appropriate for testing purposes:

```
//within the file manager...
require([], function() {
var addresses = {}, debug = true;
addresses.parse_data = function() {...}
...
if(debug === true) {
window.addresses = addresses;
}
});
```

The above example will make "addresses" globally available when the debug variable is set to true. This will allow you to do this directly in the console:

```
addresses.parse_data();
```

NOTES



For more information regarding BigMachines and the JavaScript Framework, see GS.COE.JA.18 - JavaScript Framework 2.0 Upgrade Kit.



In Firefox, ensure that you have Firebug available. In Chrome and IE, the developer tools are useful for debugging JavaScript.

You will want to use the "Script" tab in these tools, and then set break points within your code. When you reload the page, you will be able to step through the code at the point you chose.

RELATED TOPICS

USING JQUERY IN BIGMACHINES

OVERVIEW

What

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript.

Why

Adding custom JavaScript to a site at all is risky. Before adding a custom JavaScript solution make sure that you have performed a proper risk assessment. Everyone involved in the project needs to be aware of the maintenance issues that customized development causes before moving forward.



Once the decision to use JavaScript has been made, we recommend using jQuery to perform any DOM operations that need to be performed.



Proper use of jQuery can make your code more robust and safer, and also help deal with cross-browser compatibility issues.

Where

JavaScript should only be put in the BigMachines JavaScript Framework. This means that you will be writing your code in the File Manager. *jQuery 1.4.2 is available in BigMachines in the following places:*

- Home Page
- Parts Search
- Configuration
- Commerce Quotes



jQuery is NOT available on the admin side. Invoking jQuery directly from a site-wide context will cause errors in the BigMachines application.

The worst place to put JavaScript is directly in the header and footer.



Code spread across a large number of HTML/Read Only Attributes is very hard to maintain.

ADMINISTRATION

How

Before you begin

This is not a tutorial in use of jQuery, this is just a discussion of best practices when using it within BigMachines. For a tutorial on jQuery, visit the jQuery home page at: http://jQuery.com.

Learn JavaScript

jQuery is not a replacement for JavaScript, and will not protect you from many common JavaScript errors. We recommend http://e-loquentjavascript.net as a resource for learning JavaScript.

Communicate

A lot of times you will be able to do things with JavaScript that work in the immediate term, even though the solution is brittle, flimsy, and dangerous. If as an engineer you come across this, communicate it upwards as clearly as possible. Make sure that everyone involved knows exactly what the risks and issues with the solution are, and open a discussion to find an alternate solution.



Be aware that JavaScript development calls for extra rigor and attention. Take your time and do it right, custom JavaScript is time consuming and difficult enough to maintain already.

Use jQuery, not $

The $ symbol is used by a number of different libraries, and is not safe to use directly in BigMachines. If you want to, you can map the globally available jQuery to a scoped variable:

```
function($) {
//$ is safe to use here
}(jQuery);
function my_function() {
var $ = jQuery;
//$ is safe to use here
}
```

Cache jQuery objects

The following code example crawls the DOM several times to find a form and some of its inputs:

```
(function() {
function crawl() {
var action, method, country;
action = jQuery("form.my-class").attr("action");
method = jQuery("form.my-class").attr("method");
country = jQuery("input[name='country']").val();
}
}());
```

The problem with this code is that it unnecessarily crawls the DOM over and over - sometimes for the same element. A cleaner and faster way to accomplish this task is to cache the jQuery objects ahead of time:

```
(function() {
function crawl_less(){
var action, method, country, $form;
// dollar sign prefix is a convention for jQuery objects
$form = jQuery("form.my-class");
action = $form.attr("action");
method = $form.attr("method");
// passing "$form" as a parameter, this limits the search
// to the selected form, rather than the whole document
country = jQuery("input[name='country']", $form).val();
}
}());
```

Beware of complicated selectors

There are times when the only way to select an element that is nested within a table in BigMachines is to use a convoluted selector. For instance, you might see something like:

```
var $element = jQuery("table:first td.form-input:last").parent().parent().parent().parent().parent();
```

These kind of selectors are especially flimsy and dangerous to use. It is very difficult to predict what the DOM will look like in future versions of BigMachines, and code that is heavily reliant on it will be difficult to fix when the upgrade rolls around. **Take steps to mitigate this problem:**

1. Use descriptive variable names.
2. Document your intentions well with your comments.
3. Wrap the complicated selectors in their own functions.

```
(function() {
// the function name gives me a hint what we're selecting
```

```
var get_array_part_column = function() {
return jQuery("table:first td.form-input:last").parent().parent().parent().parent().parent();
}
//fires when the page is loaded
jQuery(document).ready(function() {
// cache into descriptive variable name
var $part_number = get_array_part_column();
});
}());
```

TROUBLESHOOTING

My code doesn't always run

If you find that your code intermittently fails to run, it's possible that it is actually running before the page is loaded, so your selectors don't find anything. Make sure that you are taking advantage of jQuery.ready. This is your way to guarentee that the page has been loaded entirely before running your code.

Things are happening out of order

Watch out for asynchronous events in JavaScript. *A common mistake looks like this:*

```
(function() {
function ajax_call() {
var done;
jQuery.get("http://google.com/", null, function() {
done = true;
});
return done;
}
alert(ajax_call());
}());
```

If you run this code, it will always appear as "undefined". The reason is that jQuery.get is asynchronous - it runs in a separate thread and won't be excuted in time to populate the "done" variable before alert is called. *Once you recognize this you can write your code to take advantage of it:*

```
(function() {
function ajax_call() {
jQuery.get("http://google.com/", null, function() {
when_done(true);
});
}
function when_done(value) {
```

```
alert(value);
}
ajax_call();
}());
```



Animation, AJAX, and some other jQuery functions are asynchronous and might surprise you if you expect everything to run in order.

RELATED TOPICS

ENSURE USERS ARE RUNNING LATEST JAVASCRIPT

OVERVIEW

What Is It?

When making JavaScript changes, ensure the changes are placed in a new filename. Current standard is the basename_ YYYYMMDD.js.

Browsers by default attempt to cache JavaScript and stylesheet files. As we change JavaScript to support new features, it is important that the latest JavaScript is downloaded to the users browser. Since our users can be on their own computers, we can't force a clearing of their browser cache.

By deploying new names for each changed JavaScript file the browser sees these as new files and downloads them.

How Does It Work?

The header specifies a new framework file so the browser downloads it and caches the new version. The framework specifies new names for the base JavaScript files so the browser downloads those as well. The users browser is always running your latest JavaScript code.

ADMINISTRATION

Component 1: Base JavaScript Files

For each release, copy the latest base javascript file (commerce, config, commerce_line, homepage, etc) into a new file with the basename + _ + RELEASEDATE.

Component 2: JavaScript Framework

The JavaScript framework (bm-framework.js) loads all necessary JavaScript files (commerce, config, homepage, etc). When you change one of the underlying files you need to update the framework to pull in the correct version.

Create a new version of the framework with the naming standard bm-framework-YYYYMMDD.js.

```
setup.pages = {
homepage_20130323: {
active:true
},
commerce_20130720: {
active:true
},
commerce_line_20130622: {
active:true
},
config_20130622: {
active:true
},
```

```
sitewide: {
active:true
}
};
```

Component 3: HTML Header

The HTML Header loads in the bm-framework.js. This needs to change each release to bring in the appropriate framework.

```
<script type='text/javascript' src='$BASE_PATH$/javascript/bm-framework_20130720.js' />
```

How To Use It

1. Copy latest version of base javascript file (ie. commerce_20130622.js) to new release date
2. Make appropriate changes.
3. Copy latest framework (ie. bm-framework_20130622.js) to new release date.
4. Update the setup.pages section of the new bm-framework to pull in the latest base file.

NOTES



During the release, any future changes to the base files won't automatically download since the new release date versions are cached for your development team. The development team needs to force a cache refresh. This will NOT be the case for your production users as the final version will be released at go live.

You can periodically delete older versions of your JavaScript files.



This solution uses JavaScript. This should only be adopted if you have a strong technical team.

RELATED TOPICS

Article Submitted by: Savvis

ERROR HANDLING

OVERVIEW

Errors, exceptions, and bugs are a part of building any application. By paying close attention to how we deal with and handle errors, we can create an application that is easy to maintain and administrate. By making informed and consistent decisions in our error handling, we can give end users a superior experience, while also making it easy for the next engineer who comes along to maintain our solutions.

Errors can occur anywhere in our implementations, whether in the "point and click" administration, within our scripting or custom solutions, or within external systems that we connect with as a matter of business. **Error prone operations:**

- Parsing strings into numbers
- Processing delimited strings
- Using manually entered data
- Using data from automated uploads from external systems

ADMINISTRATION

Don't Catch, Avoid

The first mistake many people make when dealing with errors is to simply hide it. In languages with exception handling built in, this looks like this:

```
//Bad example WHAT NOT TO DO
try {
somefunction();
catch(exception) {
/*do nothing!*/
}
```

The problem with this approach is that when things go wrong, it is difficult to identify and locate where the issue occurred. In some cases, this can hide the error so well that we don't even know it's happening until we notice data corruption or loss.

The preferred way to deal with exceptions is to anticipate and avoid them. We can then make informed decisions about when it is appropriate to show an error to the user, and when to fallback to default values or other responses. We want our code to fail gracefully - this doesn't always mean silently.

In fact, depending on the situation, it may be more appropriate to raise an error than to avoid it.

Throwing Errors to Validation Rules

In BML, we don't have try/catch or throw. But within Commerce, we can use the Advanced Validation rules to create our own custom errors. This is useful when dealing with large blocks of code like a pricing script, where it can be tricky to do fine-grained troubleshooting. This is a simple process:

1. Create a Main Document attribute.
2. Write to that attribute from your Advanced Modify rule.
3. Show the error using an Advanced Validation rule.

As an example, suppose that we key off of a data table for a key piece of functionality in our pricing script. We know that the data table will be updated manually by an admin, which is an error prone process. So we can show a descriptive error message if we ever notice a problem with the data in the table:

```
// Commerce Library - pricing function
//before the line item loop...
errorMessages = "";
//[...]
for line in process {
//[...]
records = BMQL("SELECT Quantity from InventoryTable");
for row in records {
quantity = get(row, "Quantity");
if(!isnumber(quantity)) {
errorMessages = errorMessages + "\nInvalid value for quantity in Inventory Table: " + quantity;
errorMessages = errorMessages + " Please contact your system administrator.";
} else {
//perform operations using data
//[...]
}
}
//[...]
}
//populate attribute
resultString = resultString + "1~customErrorMessage_quote~" + errorMessages;
```

Now in the Advanced Validation rule we simply display any error messages that came up in the Advanced Modify:

```
resultString = "";
//[...] perform any other validations
resultString = resultString + customErrorMessages_quote;
return resultString;
```

This is a nice solution in that we very clearly deal with expected problems that might come up in the life of the application. We might even take it a step further, and use workflow rules to trigger an email to the system admin with our error message. The key is that we are anticipating errors and dealing with them in a way that matches the severity of the problem.

TROUBLESHOOTING

When writing your error handling, be sure to consider the process necessary to show the error. It's not going to do you any good to create a custom error message if the Advanced Modify is unable to complete without some sort of data.

It may be necessary to provide valid default data in the event of invalid data, just so that Advanced Modify can finish running and your validation rule can fire. The advantage of this is that whoever is supporting your site will have a great idea of where to get started to fix the issue.

RELATED TOPICS

ERROR HANDLING

OVERVIEW

Errors, exceptions, and bugs are a part of building any application. By paying close attention to how we deal with and handle errors, we can create an application that is easy to maintain and administrate. By making informed and consistent decisions in our error handling, we can give end users a superior experience, while also making it easy for the next engineer who comes along to maintain our solutions.

Errors can occur anywhere in our implementations, whether in the "point and click" administration, within our scripting or custom solutions, or within external systems that we connect with as a matter of business. **Error prone operations:**

- Parsing strings into numbers
- Processing delimited strings
- Using manually entered data
- Using data from automated uploads from external systems

ADMINISTRATION

Don't Catch, Avoid

The first mistake many people make when dealing with errors is to simply hide it. In languages with exception handling built in, this looks like this:

```
//Bad example WHAT NOT TO DO
try {
somefunction();
catch(exception) {
/*do nothing!*/
}
```

The problem with this approach is that when things go wrong, it is difficult to identify and locate where the issue occurred. In some cases, this can hide the error so well that we don't even know it's happening until we notice data corruption or loss.

The preferred way to deal with exceptions is to anticipate and avoid them. We can then make informed decisions about when it is appropriate to show an error to the user, and when to fallback to default values or other responses. We want our code to fail gracefully - this doesn't always mean silently.

In fact, depending on the situation, it may be more appropriate to raise an error than to avoid it.

Throwing Errors to Validation Rules

In BML, we don't have try/catch or throw. But within Commerce, we can use the Advanced Validation rules to create our own custom errors. This is useful when dealing with large blocks of code like a pricing script, where it can be tricky to do fine-grained troubleshooting. This is a simple process:

1. Create a Main Document attribute.
2. Write to that attribute from your Advanced Modify rule.
3. Show the error using an Advanced Validation rule.

As an example, suppose that we key off of a data table for a key piece of functionality in our pricing script. We know that the data table will be updated manually by an admin, which is an error prone process. So we can show a descriptive error message if we ever notice a problem with the data in the table:

```
// Commerce Library - pricing function
//before the line item loop...
errorMessages = "";
//[...]
for line in process {
//[...]
records = BMQL("SELECT Quantity from InventoryTable");
for row in records {
quantity = get(row, "Quantity");
if(!isnumber(quantity)) {
errorMessages = errorMessages + "\nInvalid value for quantity in Inventory Table: " + quantity;
errorMessages = errorMessages + " Please contact your system administrator.";
} else {
//perform operations using data
//[...]
}
}
//[...]
}
//populate attribute
resultString = resultString + "1~customErrorMessage_quote~" + errorMessages;
```

Now in the Advanced Validation rule we simply display any error messages that came up in the Advanced Modify:

```
resultString = "";
//[...] perform any other validations
resultString = resultString + customErrorMessages_quote;
return resultString;
```

This is a nice solution in that we very clearly deal with expected problems that might come up in the life of the application. We might even take it a step further, and use workflow rules to trigger an email to the system admin with our error message. The key is that we are anticipating errors and dealing with them in a way that matches the severity of the problem.

TROUBLESHOOTING

When writing your error handling, be sure to consider the process necessary to show the error. It's not going to do you any good to create a custom error message if the Advanced Modify is unable to complete without some sort of data.

It may be necessary to provide valid default data in the event of invalid data, just so that Advanced Modify can finish running and your validation rule can fire. The advantage of this is that whoever is supporting your site will have a great idea of where to get started to fix the issue.

RELATED TOPICS

ERROR MESSAGES

OVERVIEW

Errors, exceptions, and bugs are a part of building any application. By paying close attention to how we deal with and handle errors, we can create an application that is easy to maintain and administrate. Errors can occur anywhere in our implementations, whether in the "point and click" administration, within our scripting or custom solutions, or within external systems that we connect with as a matter of business. **Error prone operations:**

- Parsing strings into numbers
- Processing delimited strings
- Using manually entered data
- Using data from automated uploads from external systems

ADMINISTRATION

Error Messages and Solutions:

The partner site could not be updated. The following errors were returned: invalid record ID.

Description: When trying to integrate with SalesForce, get error message " The partner site could not be updated. The following errors were returned: invalid record id." This error is usually caused by a recent change to the Sales Force or BigMachines set.

Solution: Check your Sales Force garbage bin, which may allow you to restore deleted content. To get a more detailed error message, which should help troubleshoot the problem further, you can check the error on the Sales Force website. To do so:

1. Navigate to the quote through the quote manager on the user-side.
2. Use your browser to **View Source**, and perform a search for "`id (double-quote, id)`".
3. Copy the value of the hidden field. *It should be a 7 or 8 digit number.*
4. Navigate from: Admin > Process Definition > Integration XSL Files.
5. Select the XSL file you are working with, and paste the ID number into the **Transaction ID** field.
6. Select **Preview SOAP XML**.
7. Copy the ID value, within `queryString`.
8. Navigate to your Sales Force page, and open up any opportunity.
9. In the URL, replace the value of ID with the value you copied from SOAP XML.

You will now be able to view the detailed error message produced by SalesForce.com.

An error occurred while communicating with the partner site.

Description: If you encounter the following message when performaning an action in BigMachines that triggers an integration with SalesForce: *An error occurred while communicating with the partner site.* You will want to check the integration associated with the action, and take a look at the SOAP Generator XSL file. Let's say we have the following code in XSL:

```
<xsl:variable name="MNamount" select="$sub_doc/relatedMaintenance/>
<Maintenance_Amount__c>
<xsl:value-of select="$MNamount"/>
</Maintenance_Amount__c>
```

If the value for relatedMaintenance does not exist in XML Document, MNamount will be undefined, and the generated SOAP XML Input will be:

```
<Maintenance_Amount__c></Maintenance_Amount__c>
```

Salesforce does not accept SOAP XML with attributes containing no value. Therefore we will receive the error "An error occurred while communicating with the partner site."

Solution: To resolve this, we just need to put a check to make sure it is defined:

```
<xsl:if test="$MNamount">
<Maintenance_Amount__c xmlns="">
<xsl:value-of select="$MNamount"/>
</Maintenance_Amount__c>
</xsl:if>
```

OR

```
<xsl:choose>
<xsl:when test="$MNamount">
<Maintenance_Amount__c xmlns="">
<xsl:value-of select="$MNamount"/>
</Maintenance_Amount__c>
</xsl:when>
<xsl:otherwise>
<Maintenance_Amount__c xmlns="">
<!-- Default value -->
</Maintenance_Amount__c>
</xsl:otherwise>
<xsl:choose>
```

Page cannot be displayed.

While using the BigMachines application in Internet Explorer, trying to save a quote or add line lines "Page Cannot be Displayed" error will display. This is due to settings that need to be set in the Internet Explorer browser. Please see Recommended Browser Settings for details on browser settings. For more information, see Browser Compatibility.

Error printing output: FO to RTF/PDF

Description: An error occurs when printing output - FO to RTF or FO to PDF. This usually occurs when a change has been made to any custom text that is displayed on the printed output, or else a change to the Doc Engine. It may also happen on sites that still use the Proposal Editor, which is a legacy version of the Doc Engine.

Solution: The first step is to track back through any recent changes you have made, and verify that there aren't any invalid characters or incorrect HTML formatting in the edited section. Remember to never paste from Microsoft Word into BigMachines fields, and if you suspect that someone has, remove the offending section completely and reenter it, using the controls on the browser text editor to format the text. *If you still cannot find the source of this error, follow these steps:*

1. Open the problem quote in the Quote Manager.
2. View Source, and locate the `BS_ID` by searching for "`id (open quote, id).`
3. Navigate from: Admin > Process Definition > Printer Friendly History/XSL Views.
4. Enter the appropriate XSL section. *If you're not sure which it is, you can look under the Action Admin for the button you pressed to see the error, under the XSL tab.*
5. Paste the `BS_ID` into the **Transaction** field.
6. Choose **No Converter** from the **Process Output Through** drop-down menu.
7. Click **Apply**.
8. When the page stops loading, select **Preview Document**. This spits out a pure XSL file now, because you turned off the converter.
9. Make sure you are viewing the XSL file in IE. *If you're in Firefox, just save it to the desktop and view it in IE.*
10. Scroll all the way to the bottom of the page. *There should be an XML error, and it will be right next to some text that should help you identify where exactly the problem is happening.*



Remember to turn the converter back on in the "Process Output Through," or the next deploy will cause the users to see the bare XSL.



Never copy/paste directly from Microsoft Word into BigMachines fields.

User does not have permission to view this page.

Description: User get error that they do not have permission to view a page. This error occurs when a user does not have the User Access Rights in a specific step of a quote.

Solution: The User Access Rights are managed under Admin > Process Definition > Steps > {Step Name} > Participant Profiles > User Access Rights. You may also need to check which group the user is in, by navigating to Admin > Users > {UserName} > Groups

tab. The user will be placed into the first Participant Profile, from top to bottom, which has granted him access. For more information, see Participant Profiles.

Null Pointer Exceptions

What: A null exception error occurs when you refer to an array index that is not populated. It will show an error to the users, and appear in the application error logs.

Why: Null pointer exceptions are common, so dealing with and avoiding them will save a lot of time and avoid bad user experiences.

Where: Any BML function.

How: Avoiding a null-pointer exception is simple and straight-forward; test that the array index exists before accessing it. *You can do this with the* `sizeofarray` *function.* You should also do this when working with dictionaries, even though the dictionary will not return the null pointer error. *The* `containskey` *function will allow you to check for the existence of a dictionary element before accessing it.*

Example:

```
result = "";
delimiter="!_-_!";
delimited_string = "discount_value"+delimiter+"10.0"+delimiter+"discount_percent"+delimiter+"0.5";
//parsed_array will only contain 4 items
parsed_array = split(delimiter, delimited_string);
//test for length before accessing!
if(sizeofarray(parsed_array) > 10) {
result = parsed_array[10];
} else {
// deal with unexpected data transparently, don't just hide it
result = "Error parsing string in util library: " + delimited_string;
}
return result;
```



The error logs will show the exception, along with a line number. Be prepared to locate the error in Util libraries. *This means building a list of parameters that will be passed to the function, and may involve writing a test script to debug all the way to the final line.*



Users who are not full access will likely see the "Contact your system administrator" message.

LEGACY

HTTP 500 Response Code while calling Update Transaction

This is an internal server error. Occurs in Version 10 and 11 sites, the `updateTransaction` SOAP call has different mandatory attributes and changes to the format of input data.

Confirmed mandatory attributes for bm:quote_process:

```
<bm:quote_process bm:data_type="0" bm:document_number="1">
```

Failure to include these attributes will give an error of the following kind in the Error Log:

```
15 Apr 2011 14:11:11,201 [ERROR] com.bm.xchange.util.UrlEx.sendDataToHtmlUrl(UrlEx.java:238) - Could open connection.
java.io.IOException: Server returned HTTP response code: 500 for URL: https://TestIn-formatica.bigmachines.com/v1_0/receiver
```

WebServices test error:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<soapenv:Fault>
<faultcode>soapenv:Server</faultcode>
<faultstring>COMMERCE-ERR:For input string: ""</faultstring>
<detail xmlns:bm="urn:soap.bigmachines.com">
<bm:fault>
<bm:exceptionCode>COMMERCE-ERR</bm:exceptionCode>
<bm:exceptionMessage>For input string: ""</bm:exceptionMessage>
</bm:fault>
</detail>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>
```

Some non-string data types can no longer be enclosed in CDATA tags

This will cause the parser to interpret the CDATA tags as part of the attribute and throw an error. This has been confirmed for integer and date data types. *Adhere to this format for Integers and Dates:*

```
<bm:adHocQuantity2_quote>22</bm:adHocQuantity2_quote>
<bm:expirationDate_quote>2011-03-22 00:00:00</bm:expirationDate_quote>
```

To be safe, do not leave any white space characters, newlines, carriage returns or tabs between the attribute value and the enclosing XML tags.

Namespace for the <data_xml> tag is now required

Old Format *(successfully processed by WebService)*:

```
<bm:updateTransaction xmlns:bm="urn:soap.bigmachines.com">
<bm:transaction>
<data_xml>
<bm:quote_process bm:data_type='0' bm:document_number='1'>
<bm:sub_documents>
<bm:line_process bm:data_type='3' bm:document_number='2'/>
<bm:line_process bm:data_type='3' bm:document_number='3'/>
```

New Format *(old format will throw the above error)*:

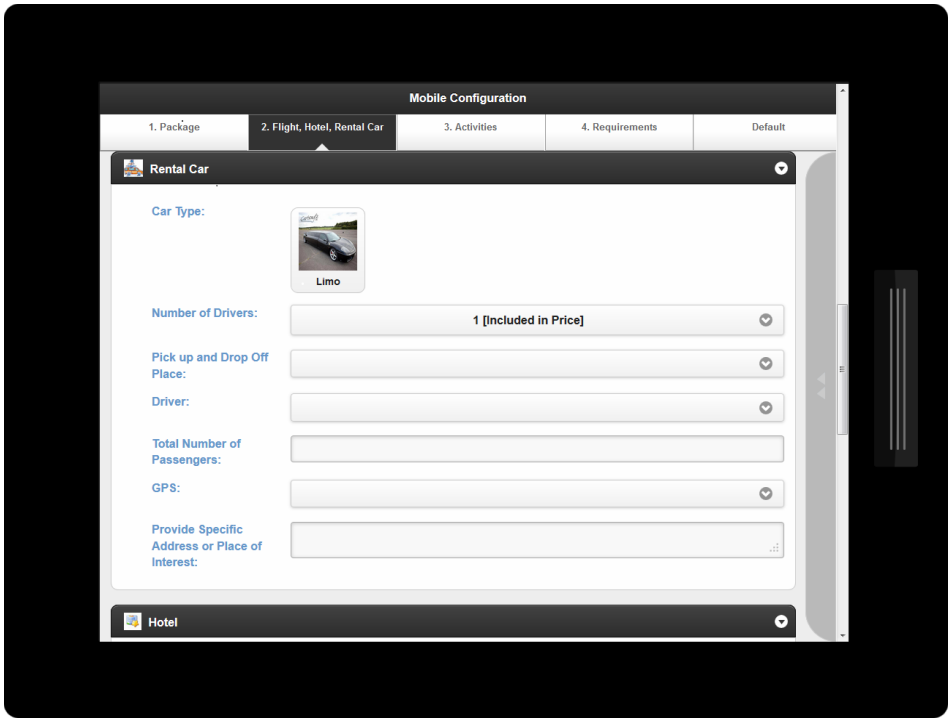
```
<bm:updateTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="h">
<bm:transaction>
<bm:data_xml>
<bm:quote_process bm:data_type="0" bm:document_number="1">
```

RELATED TOPICS

MOBILE OVERVIEW

OVERVIEW

BigMachines browser-based solution for Mobile offers customers: better security control, lower maintenance cost, and improved user experience. Mobile users can present complete quotes to a new prospect, configure quotes, and create/edit quotes on a pre-configured or newly configured item. Mobile will benefit the sales users, who are out in the field, the most. It improves win rate, reduces sales cycle time, and involves the end customer in a collaborative quoting experience. Organizations that leverage collaborative selling on tablets also reduce their sales expenses and differentiate themselves among their competition.



Mobile functionality

- Mobile functionality includes: Punch-Ins, Pending Configurations, and Parts Search. Shopping Cart and Return To Opportunity have been excluded.
- Parts Search and Parts Search Results have a Mobile optimized page.
- For optimum performance, it is recommended that Landscape Orientation is used.
- User is automatically redirected to Desktop Version of page when Mobile Version does not exist or user does not have a Mobile license.
- Mobile Navigation button, which appears in the top left corner, contains links from Commerce and Subheader sections, depending on the user type. *The admin can set permissions here: Navigation Menus > List Links > Customize Menus.*

NOTES



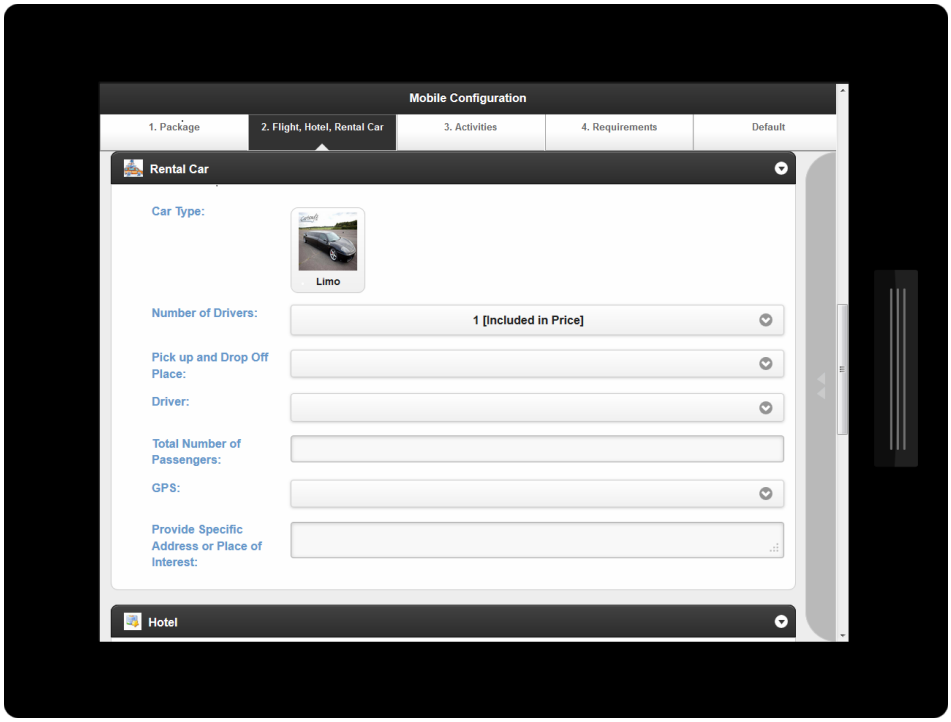
BigMachines Mobile functions in the same manner as the Desktop version. *Please refer to the standard Desktop Help articles for more information on existing functionality*.

RELATED TOPICS

MOBILE OVERVIEW

OVERVIEW

BigMachines browser-based solution for Mobile offers customers: better security control, lower maintenance cost, and improved user experience. Mobile users can present complete quotes to a new prospect, configure quotes, and create/edit quotes on a pre-configured or newly configured item. Mobile will benefit the sales users, who are out in the field, the most. It improves win rate, reduces sales cycle time, and involves the end customer in a collaborative quoting experience. Organizations that leverage collaborative selling on tablets also reduce their sales expenses and differentiate themselves among their competition.



Mobile functionality

- Mobile functionality includes: Punch-Ins, Pending Configurations, and Parts Search. Shopping Cart and Return To Opportunity have been excluded.
- Parts Search and Parts Search Results have a Mobile optimized page.
- For optimum performance, it is recommended that Landscape Orientation is used.
- User is automatically redirected to Desktop Version of page when Mobile Version does not exist or user does not have a Mobile license.
- Mobile Navigation button, which appears in the top left corner, contains links from Commerce and Subheader sections, depending on the user type. *The admin can set permissions here: Navigation Menus > List Links > Customize Menus.*

NOTES



BigMachines Mobile functions in the same manner as the Desktop version. *Please refer to the standard Desktop Help articles for more information on existing functionality*.

RELATED TOPICS

MOBILE CONFIGURATION

OVERVIEW

Layouts are automatically configured to recognize that user is on a mobile device. *Mobile Configuration includes:*

- Mobile layout support
- Utilize device specific field display
- Optimize page load and design

ADMINISTRATION

User side

- User can scroll through lists using vertical swiping and perform horizontal swiping on tabs and array attributes.
    - The arrays have only one array-type vertical scaling option whether the array is tabbed or horizontal.
- On-demand tab loading: only the content of tab that is being viewed is loaded.
    - Tabs are located in the top panel.
    - If there are more than five tabs on the page, only the first five are displayed; the user can swipe to see the other tabs or use the integrated list menu.
    - A button at the bottom of each tab allows users to navigate to the next tab.
- Array pagination: 10 array indices per page. *Rules will run on all indices, each index that contains a message will be highlighted and only the selected index messages will be displayed to eliminate vertical scrolling.*
- Action bar at bottom of the tablet page contains the following buttons, when applicable: Next, Previous, Start Over, Save, Cancel, Add to Quote, Return to Quote, Create Quote, and Add to Favorites.
    - Mimics existing Desktop functionality.
    - Contains Pipeline Viewer for FullAccess user types.
    - Action bar displays up to five buttons.
        -  "More" button contains any additional buttons.
    - Scenarios in which the action bar is used:
        - **Next and Previous:** used in multi-node Config flows.
        - **Save and Cancel:** used in Reconfiguration contexts.

- **Create Quote:** only used when a Commerce transaction has not been created prior to Config. *This action is temporarily hidden until "Pending Deletes" is undone or an update to "Delete" is performed.*
- **Update:** saves current values and runs rules again with the current Config values.



Any additional Invocation buttons will be added to the action bar.

- All Configuration rules will fire on Mobile as they would on Desktop. *Any attribute that is on the Desktop layout, but not on mobile layout behaves as if it is hidden on Mobile.*
  - All attributes that are on the Desktop layout will be included in the Pipeline Viewer.
  - If a constraint rule fires on a "hidden" attributes, an internal constraint error will occur.



Recommended Items and Pricing Sections appear in the collapsible sidebar on the right hand side.

User can scroll through lists using vertical swiping and perform horizontal swiping on tabs and array attributes. *The arrays have only one array-type vertical scaling option to matter if the array is tabbed or horizontal.*



Layout is organized into a one column to reduce the need for horizontal scrolling.

Mobile functionality takes advantage of the native functionality of a tablet device. **Examples:** Date Picker and drop-down menus.

When AJAX-enabled attributes are changed, AJAX rules run and the results display as expected. *Pricing attributes will update the pricing information.*



Array Control Attributes must be located on the same tab as the Array Set they control.

Admin side

- Mobile layouts are created within the Flow Properties page.
    - Mobile layouts are initially generated to contain the contents of the Desktop layout.
        - The administrator may then edit Mobile layouts to contain fewer attributes than that of the Desktop layout. However, the Mobile layout cannot contain attributes not on the Desktop layout.
    - In order to regenerate the Mobile layout, the admin must delete the Mobile layout and restart the “Create Mobile Layout” process.
- Array sets can be dragged-and-dropped from the sidebar into groups.
    - Array sets contain a Delete button, but no Edit button.
    - Attributes within the sets contain Edit buttons, but no Delete buttons.
        - Array sets are ordered according to the associated set order.
- The admin Mobile layout will be auto-generated from the Desktop layout the first time it is created.
    - Once a Mobile layout is created it becomes a separate layout that must be updated independently in order for user side changes to take effect.
        - The only attributes available to a Mobile layout will be the attributes which are already present on the desktop lay-out.
    - When an attribute exists on Desktop, but not on Mobile, the attribute is still present, just hidden, on the mobile layout.
        - Any rules which are related to that attribute will still fire when users access the Mobile layout.
        - If a required field is not added to the mobile layout and it lacks a default value, the user will be unable to create a quote.
- “Mobile Device Enabled” system attribute is now available as a rule input for Recommendations, Constraint, and Hiding rules.
    - Attribute will return “True” if the following conditions have been satisfied:
        - User is on Mobile device.
        - Administrator has enabled Mobile access for user.
        - Administrator has defined Mobile layout.
    - System attribute returns “False” if one or more of the above conditions have not been met.



Attributes can be moved, edited, and deleted on the Mobile layout.

An alternate stylesheet can be uploaded for Mobile. *All of Mobile utilizes one alternate stylesheet.*



Action bar at bottom of tablet contains the following buttons: Remove Layout, Layout Properties, and Preview.



Multiple column layouts on Desktop will be converted into a single column on Mobile.

Layout Properties has been moved from the Config Flow Editor page to the Desktop and Mobile Layout Page.

Custom JavaScript revisions

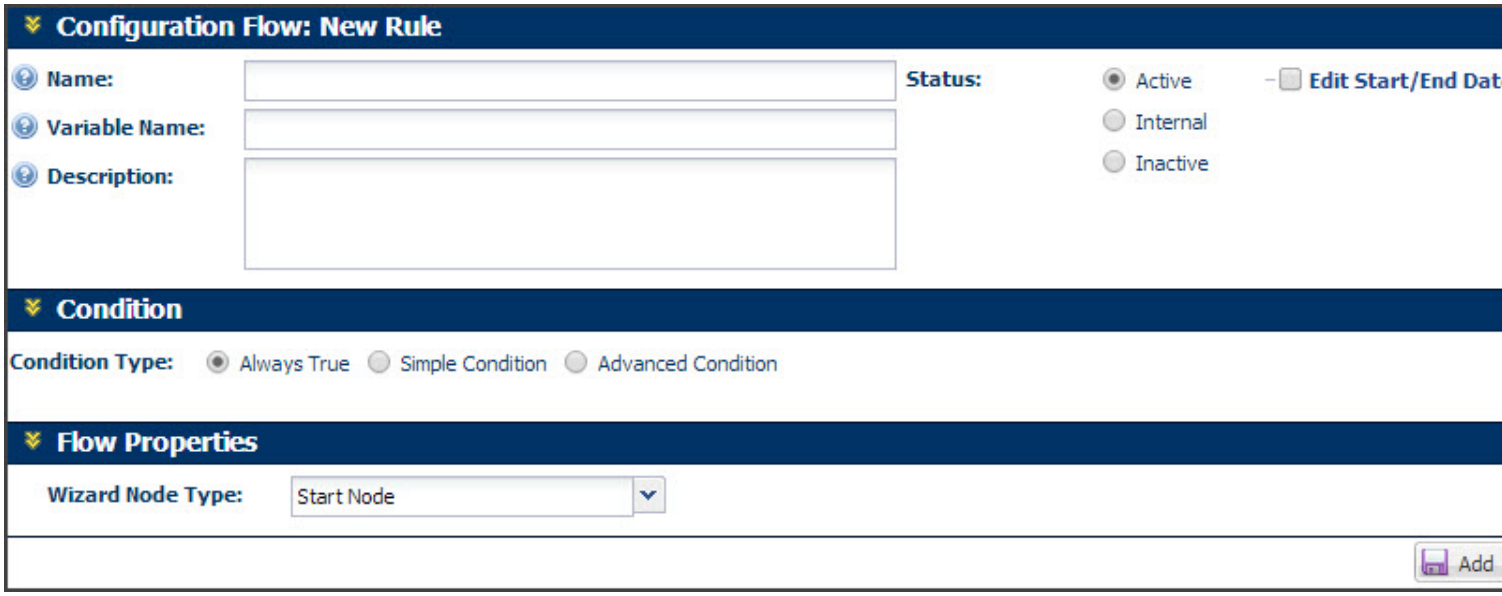
Sites using any custom JavaScript must revisit the areas in which custom JavaScript has been implemented in order for Mobile layouts to appear as expected. *Custom JS is usually implemented in the following areas:*

- Headers
- Footers
- JS files uploaded to the File Manager through Config or Commerce HTML Attributes

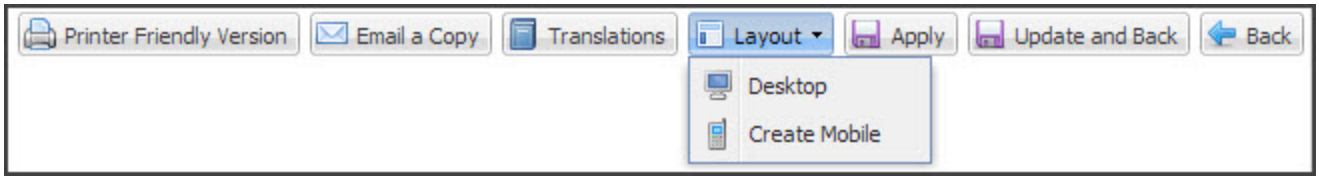
---

Creating a Mobile Configuration Flow

1. Navigate from Admin Home Page > Products > Catalog Definition > All Product Families > List > Select Product Family > Configuration Flow
2. Click **Add** to create a new flow rule.
3. Enter the **Name**, **Variable Name**, and **Description** of your rule.

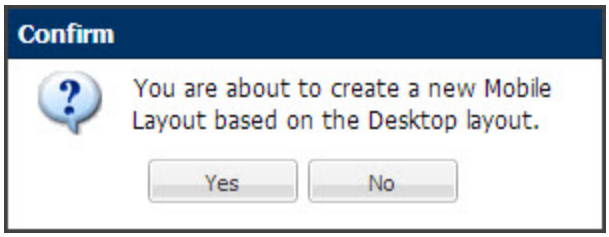


4. Select the **Status** of the rule.
    - **Active:** The rule must fire if the condition is satisfied.
    - **Inactive:** The rule does not fire, even if the condition evaluates to true.
    - **Internal:** Only fires if the FullAccessUser is the user and the condition evaluates to true.
5. If applicable, click **Edit Start/End Dates** to select the time period for which the rule is valid. *If no Start/End dates are selected, then it is assumed that the status is valid until the fullaccessuser changes it.*
6. Choose your **Condition Type.**
7. Specify the **Wizard Node Type:**
    - **Start Node:** Where the config flow will begin. You can have more than one start node, so if you plan on having multiple, this will be what you set your beginning nodes to.
    - **End Node:** Final node of the flow. This is where you'll see your action buttons like Add to Quote, as well as any recommended line items. As with start nodes, you can have multiple end nodes. The proper end node is set by attributes in the condition.
    - **Start and End Node:** Start and End Nodes are combined.
    - **Transition Node:** Transitions user from one node to another based on attribute selections.
    - **Primary Start Node:** When there is more than one start flow, this lets the application know to start here first.
    - **Primary Start and End Node:** Primary Start and End nodes are combined.
8. Click **Add**. *The following buttons will appear below the newly created Configuration Flow Rule:*

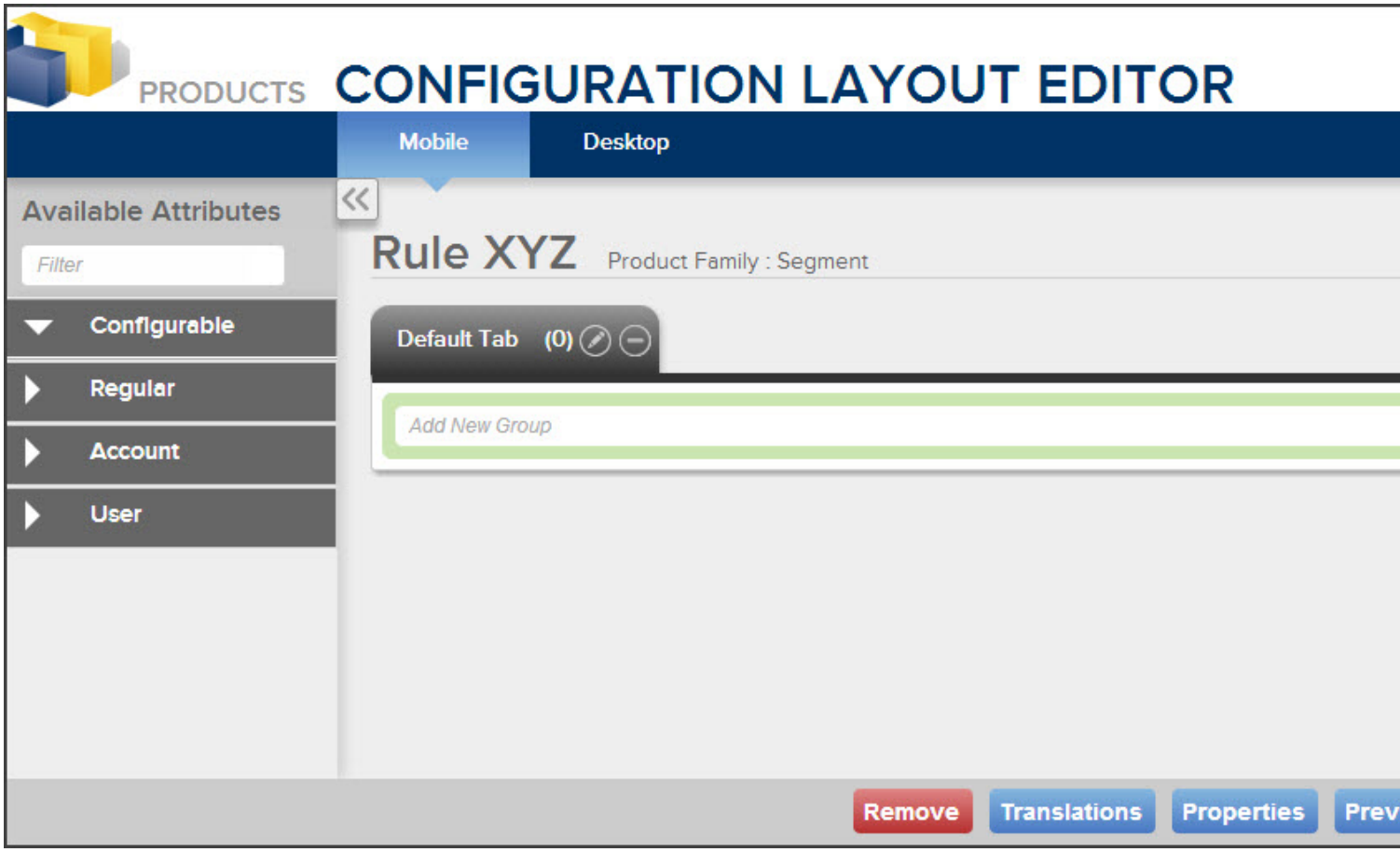


9. Click **Layout**.

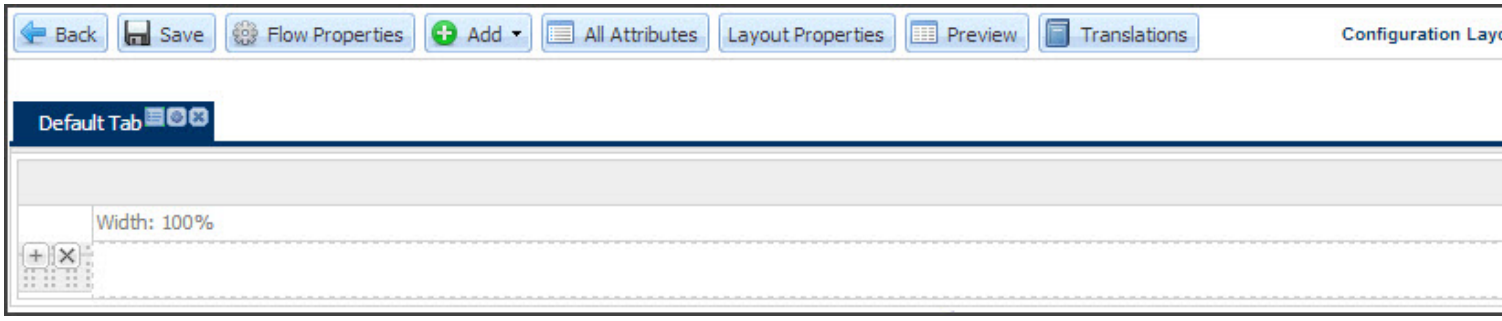
10. Select **Create Mobile**.The following confirmation dialog will appear: *For more information on Configuration Flows and/or creating a Desktop Configuration Flow Rules, see Configuration Flows.*



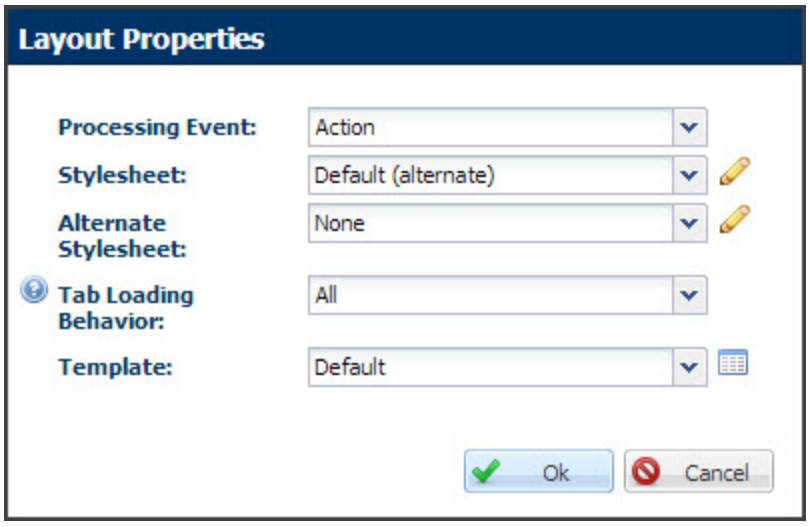
11. Edit **Configurable**, **Regular**, **Account**, and **User Attributes** on the Mobile Configuration Layout Editor. *This editor functions in the same manner as the Email Template Editor; for more information, see Email Templates Overview.*



12. Select the **Desktop** tab to make changes to the Desktop Configuration Flow. *This will direct you to the following page:*



13. Click **Layout Properties**. The following dialog will appear:



14. Specify the **Processing Event**:
    - **Action:** The condition for the flow is evaluated when an action occurs (the user clicks a button) on the Model Configuration page.
    - **Value Change:** The condition for the flow is evaluated when the users change a value for a configurable attribute.
    - **Tab Change:** The condition for the flow is evaluated when the users move from one tab to another on the Model Configuration page.
15. Select a **Stylesheet** from the drop-down menu. Click the pencil icon to make changes to the stylesheet.
16. Select an **Alternate Stylesheet** from the drop-down menu. Click  to make changes to the stylesheet.
17. Select **Tab Loading Behavior**:
    - **Load all tabs on page load:** Loads all data across tabs.
    - **Load only active tab on page load:** Loads data on only the active tab in configuration. Note: Users might see a loading page appear when the switch tabs, as the next set of data is being loaded. Select this option improves performance and rendering speed on larger configuration flows.

18. Select a **Template.** Click  to preview the template.
19. Click **Ok**. *This will close the **Layout Properties** dialog.*
20. Click **Save**.
21. Click **Add** to begin setting up the user-side layout.



Desktop and Mobile Layouts are considered separate Flows. If a change is made to one flow, it will not automatically be reflected in the other. User must make changes in BOTH the Desktop and Mobile Layout.

NOTES



Although mobile layout is visible in iFrame, optimal user experience is achieved through the full window. The "Occupy entire window when the site is inside a frame" parameter controls the iFrame setting for both Desktop and Mobile, located in General Site Options. *To control this parameter user must select "True" to occupy the entire window when the site is inside a frame.*



BigMachines Mobile functions in the same manner as the Desktop version. *Please refer to the standard Desktop Help articles for more information on existing functionality.*

Pages with a large number of attributes or array rows could see performance issues. *It is highly recommended that users break up large sets of data to mitigate potential performance issues.*



Migrations will fail if the destination site and source site's Mobile Enabled settings are not identical.

Configuration calculators are not supported on Mobile.

RELATED TOPICS

MOBILE HOME PAGE

OVERVIEW

Mobile Home Page mimics the functionality of the desktop Home Page, but utilizes finger interaction, “gestures”, instead of typical desktop mouse functionality. Administrators can set permissions for the Mobile Home Page.

ADMINISTRATION

User side

- Users can swipe horizontally through Product Family tabs.
- Pending Configuration has been optimized for Mobile display; they only appear when user has Pending Configs.
    - The Pending Configs page can be accessed by swiping from the Model Punch-In page or on the Navigation link at the bottom.
- Parts Search and Parts Search Results has been optimized for Mobile. *Part Search performs a simple search of the user-entered String against Part Descriptions and custom fields.*
    - Part Search displays the first 25 results. *Click the **More** icon to see the next 25 results.*
        - Parts selected from Parts search will continue to be selected/accumulate, even if another search is performed, until user returns to the quote or Commerce.

Admin side

- “Mobile Device Enabled” system attribute is now available as a rule input for Home Page permissions.
    - Attribute will return “True” if the following conditions have been satisfied:
        - User is on Mobile device.
        - Administrator has enabled Mobile access for user.
    - Will return “False” if any of the above conditions have not been met.
- Model Punch-In descriptions are pulled from the text field in the Model Properties page.
- Mobile Home Page display includes Model Punch-In selections with labels, descriptions, and images.
    - Images can be added to the Mobile Punch-In using the image Browse/Upload function, located on the Admin Home Page.
        - Recommended maximum image width and height: 235px by 150px.

NOTES



Recommended maximum image width & height: 235px by 150px.



BigMachines Mobile functions in the same manner as the Desktop version. *Please refer to the standard Desktop Help articles for more information on existing functionality.*

Attribute range calculations are validated on the Value Punch-In. **Example:** if the value was passed in the URL to the Configuration, it will be validated and an error message will appear. *This affects both Mobile and Desktop.*

RELATED TOPICS

MOBILE APPROVALS

OVERVIEW

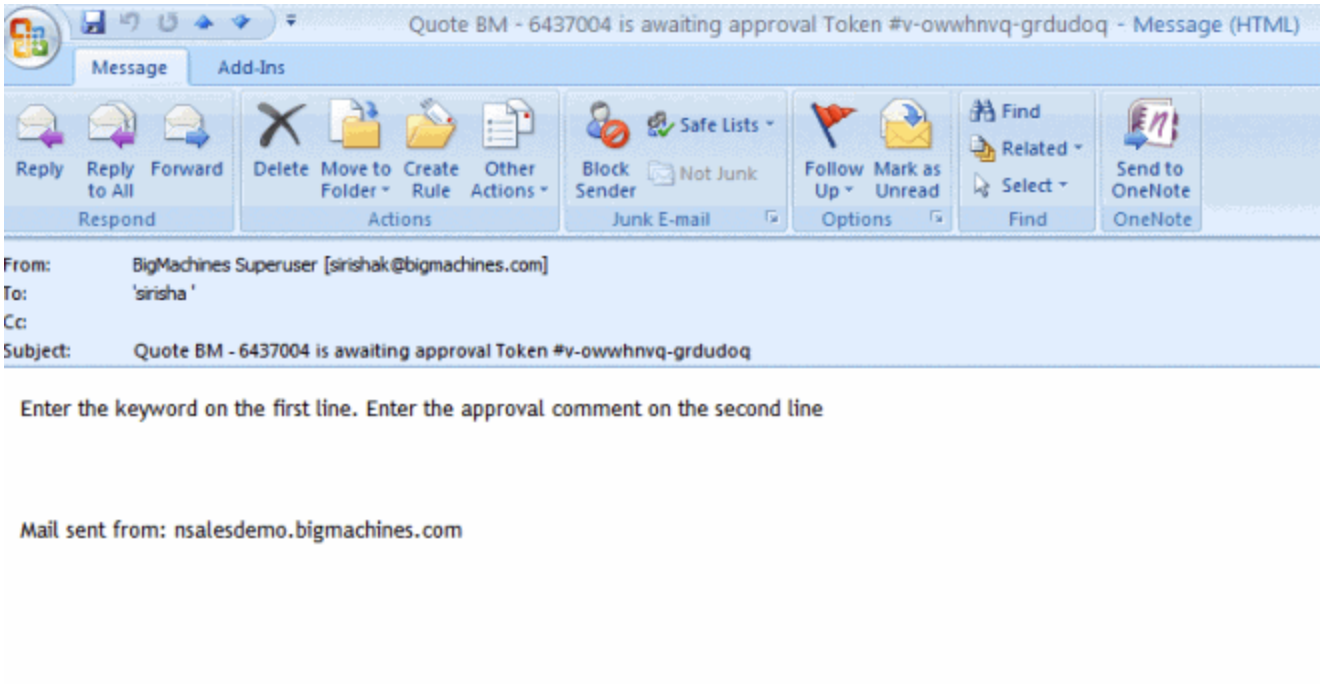
What are they?

This feature allows users to perform modify actions on a quote by replying to an email. The typical use case for this is to allow a user to approve/reject a quote by responding to a system generated email from a mobile device. The user can also provide a comment in the email that can be mapped to a history or text area type attribute. For more information on this functionality, watch this demo.

How do they work?

When a BigMachines registered user performs a step change action then a notification is sent to the user. If the Mobile Devices approvals process is set up on this notification rule, then the email sent to the user will contain a unique key in the subject line.

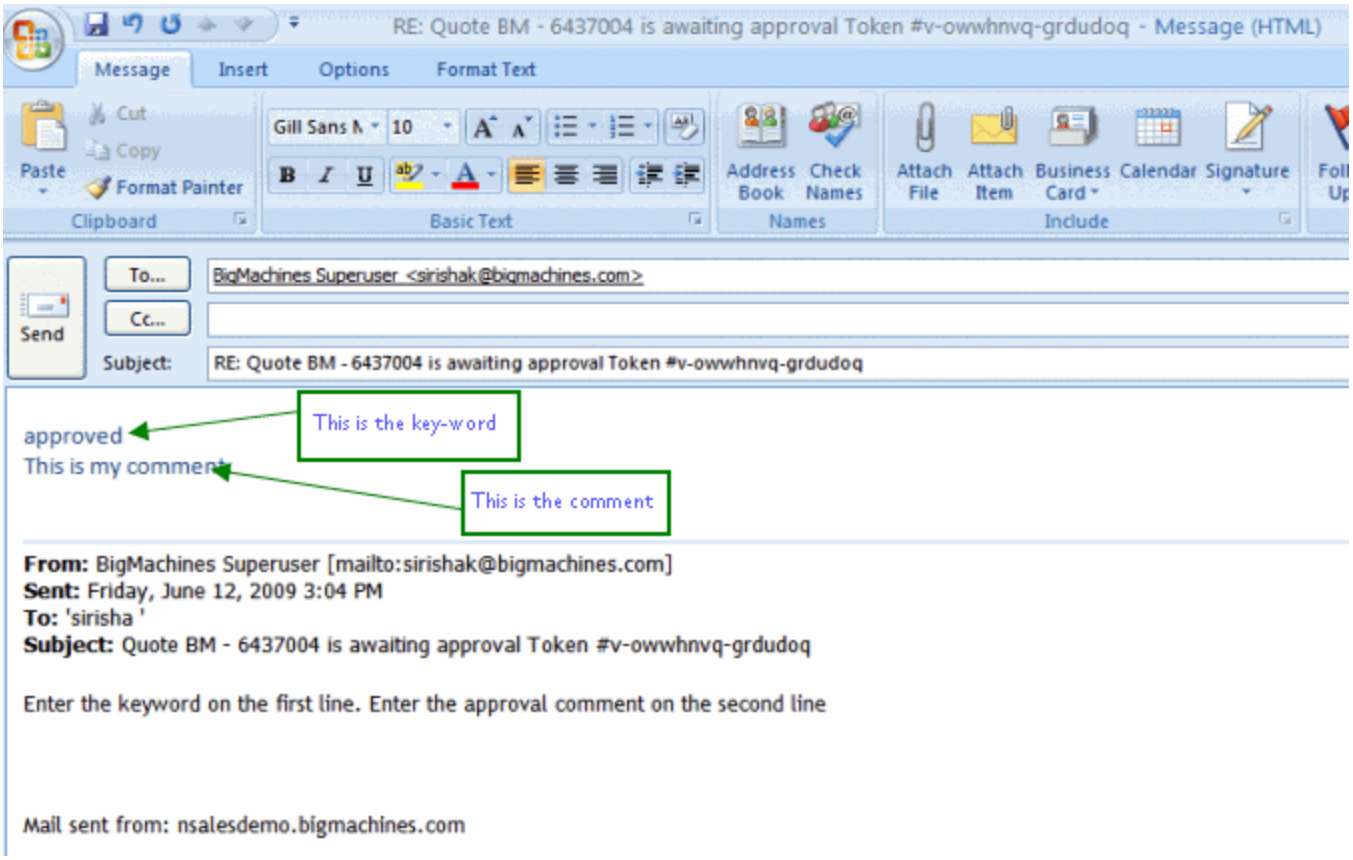
Sample Email with Unique Key



The approver will need to reply to the notification email to perform the modify action. **The subject line of the email must contain the unique key**.

- The first string on the first line of the email will be the keyword associated with the action that the user is trying to perform.
- The second line of the email can be the comment. This comment can be mapped to any text area or history type attribute in commerce.

Sample Reply to the Notification Email



Then, BigMachines sends an email indicating if the action specified by the user was performed successfully or if an error was encountered.

ADMINISTRATION

Setting up Mobile Approvals

1. Navigate to a **modify-type** action. For example, "Approve".
2. Define the **Email Notification keyword** for the action.

   Keywords are not case-sensitive.

3. Multiple keywords can be associated with an action. All possible values can be separated with a tilde in the Email Notification Keyword field on the Action Editor.

- A keyword has to be unique.
- A keyword can have alphabets, numbers and underscore.
- A keyword cannot have multi-byte characters
- A keyword can be associated with only one action within a process

4. Specify a text area or history type attribute in the Approval Comment Mapping field in the Action Editor.
5. The comment specified by a user in the email will be mapped to the specified attribute.
6. On a Transition Rule,set up the Mobile Devices Notification. Select the option Allow response to Trigger an Action.
7. When this option is selected, BigMachines will automatically generate the unique key in the subject line of the outgoing email.

NOTES



The user specified comment can contain multi-byte characters.



BigMachines validates that the user email is registered with the system before sending the unique key. Therefore, only a user with a valid BigMachines account can receive the unique key and can perform the modify action via email. Inactivated users will not be sent a key.

The user responding to the email is used for permissions on actions and any CRM integrations.



The person replying to an approval email must have the same email address as the person to whom the approval notification was sent.

RELATED TOPICS

BigMachines, Inc.
570 Lake Cook Road Suite 200
Deerfield, IL 60015

RELEASE NOTES

The BigMachines 2013 Autumn Release Notes provide high-level details of features and enhancements implemented in this release. This document also provides information on specific upgrade requirements from older versions of the BigMachines application.



Last Updated: December 20, 2013

© Copyright 2013 BigMachines, Inc. All rights reserved. Confidential and Proprietary Information of BigMachines, Inc. No reproduction or distribution is permitted without the express written consent of BigMachines, Inc.

CONTENT

Introduction to BigMachines 2013 Autumn 4

Migration Management

Migration Rollback & Enhancements 5

Summary 5
Administration Overview 5
Notes 8

Formula Management

Formula Management Enhancements 9

Summary 9
Administration Overview 9
Notes 11

Functions

Function to Function Calls 12

Summary 12
Administration Overview 12

Mobile

Mobile Configuration 15

Summary 15
Administration Overview 15
Notes 16

Mobile Home Page 18

Summary 18
Administration Overview 18

Help System

Help Site 19

Summary 19
Administration Overview 19

General Enhancements

Secure File Manager 20

Summary 20
Administration Overview 20
Notes 20

Line Item Performance Enhancements 22

Summary 22

Administration Overview 22
Oracle Fusion 23
Summary 23
Administration Overview 23
Notes 23
Translations Bundle 24
Summary 24
Administration Overview 24

Additional Enhancements
Login allows periods 25
Price Book columns increased to accommodate more currencies 25
Mass Update function can be performed on a subset of steps 25
BMQL: return Part Description in specified language translation 25
Reconfigure in a different language 25
BML function keys() added 25
"Web Services Only" user permissions checkbox 25
Export Attachment action language pop-up 26
Enhanced Security 26
Third Party Software 26

Resolved Issues
Users/Groups/Security 27
Commerce 28
Configuration 31
CRM Integration 32
Parts 32
Pricing 33
Doc Engine 33
Reporting 33
Web Services 34
BML 34
Currency 34
Bulk Upload/Download 34
Data Tables 35
Formula Management 35
Internationalization 35
File Manager 36
General 36

Upgrading to BigMachines 2013 Autumn 37
Browser Support 37
Training 37
Disclaimer 37
Unplugged 37

INTRODUCTION TO BIGMACHINES 2013 AUTUMN

The BigMachines 2013 Autumn release notes contain highlights of all major new features and general enhancements in release 2013 Autumn. For every new feature, the Release Notes provide:

- A high-level description of the functionality
- Important notes on any deviations in functionality from prior releases
- Best Practices for implementation
- Upgrade Impact

Note: The Release Notes do not cover all the details of new features and it is recommended that you read the product help for more information.

NEW FEATURES IN BIGMACHINES 2013 AUTUMN

FEATURE	DESCRIPTION
Migration Rollback & Enhancements	Enhances existing functionality introduced in Version 12. Snapshots allow for Rollback to a previous state of migrate-able data.
Formula Management Enhancements	Enhancements allow admins to address key pricing scenarios that previously required BML.
Function to Function Calls	Commerce Library functions can call other Commerce Library functions. Util functions can call other Util functions.
Mobile Configuration	Facilitates collaborative selling by presenting sales users, who configure on tablet devices, with distinct tablet-oriented layouts and user experience.
Mobile Home Page	Distinct Home Page optimized for tablet users.
Help Site	New Help site features improved navigation and interactivity for optimal usability.
Secure File Manager	Secures files at the folder level. Only users logged into a BigMachines session may access secured files within the File Manager.
Line Item Performance Enhancements	Scales up to manage large volumes of line items faster.
Oracle Fusion	Integration with CRM system, Oracle Fusion. This allows for SSO capabilities and communication with a variety of Fusion objects.

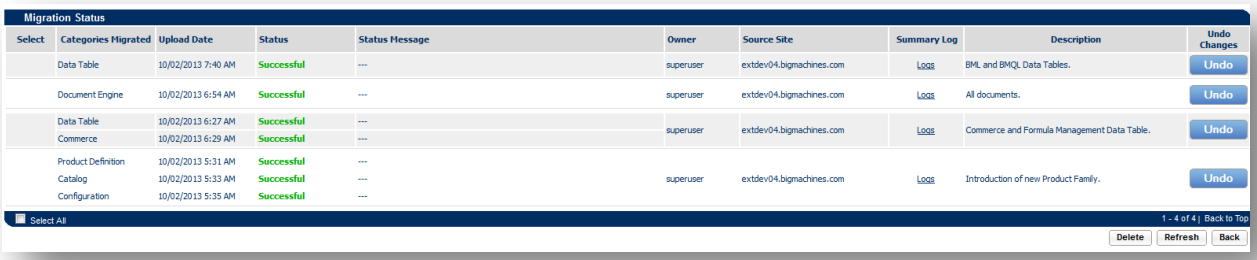
MIGRATION MANAGEMENT

MIGRATION ROLLBACK & ENHANCEMENTS

- Rollback: Undo changes on a target site that were previously migrated; returns a target site to a stable state.
- General enhancements to migration process: Data Columns can be deleted in migration, description field for each migration, and improved error messages surrounding BML compilation issues.

SUMMARY

BigMachines has improved upon the Migration Center since its introduction in Version 12. Now, if a user migrates something unwanted or experiences unexpected behavior, the target site can be Rolled back to a previous state, known as a "snapshot". Other enhancements include: Data Column deletion on a target site through migration, description field that can be populated at the time of migration, improved error messages, and resolved image path issues during migration.



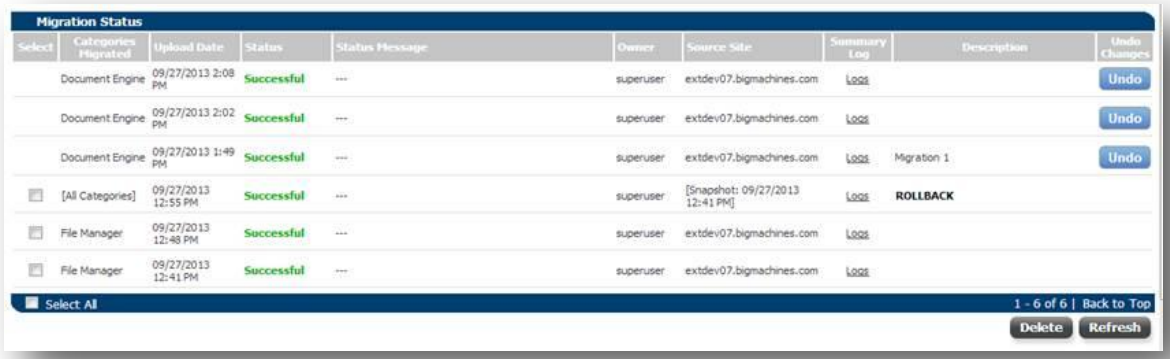
ADMINISTRATION OVERVIEW

Rollback:

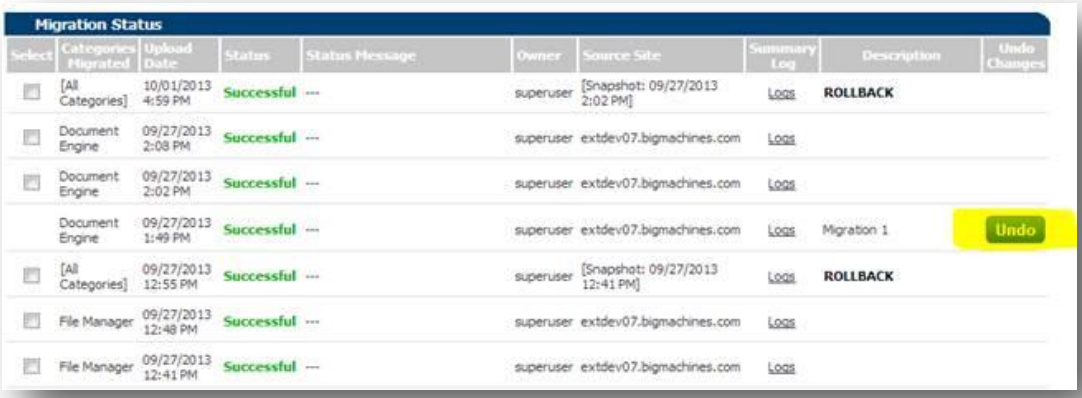
General Behavior

- Rollback is a permanent action. Rollbacks cannot be undone.
- When an Admin initializes a migration, the system automatically stores a "snapshot" of the destination site prior to performing the migration.
  - Snapshots include all migrate-able categories, with the exception of Data Tables and File Manager files that were not selected for that respective migration.
    - Snapshots capture only the deployed version of each category.
  - Within the Migration Status page, records for successful migrations will have an inline "Undo" button. Each "Undo" button is associated to the respective snapshot taken immediately prior to its inline migration.
    - Upon selecting an "Undo" button, the target site will revert back to the state captured by the associated snapshot, i.e. the state immediately prior to that migration.
      - Records for Rollbacks will also be included within the Migration Status page.
  - Migration records with the "Undo" action available cannot be deleted.

Before Snapshot



After Snapshot



- The system will store up to 10 snapshots. Accordingly, the Migration Status page will contain up to 10 migration records with an inline "Undo" button.
  - Snapshots are stored for successful migrations only. The snapshot taken immediately prior to a migration will not be stored if the migration is unsuccessful.
  - If the number of snapshots attempting to be stored reaches 11 and, accordingly, there would be 11 migration records with an "Undo" action present, the oldest snapshot will be permanently deleted from the database and the 10 most recent snapshots will remain. This deletion will occur upon completion of the next successful migration.
  - Once a snapshot is selected for Rollback, i.e. "Undo" has been selected for the inline migration record, that snapshot and all other snapshots that were taken after the selected snapshot will be permanently deleted.
- Rollback will not delete Steps.
  - If a Step exists on the target site in current state but does not exist in the snapshot selected to roll back to, the Step will remain on the target site with the following effects:
    - All Transactions in the Step will remain untouched by the system.
    - Transition Rules and Forwarding Rules under the Step will remain untouched. Note that this will allow any transaction in the Step to still transition out as it would have at the time Rollback was initiated.
    - Transition Rules and Forwarding Rules for all other Steps will be reverted to the state of the snapshot. Note that because the Step did not exist at the state of the snapshot and thus no rule could have existed to allow a transition into the Step, transactions will no longer transition into the Step after Rollback without direct Admin changes or an additional migration of Commerce.

        - All Participant Profiles, User Access Rights, and Document Views for the Step will remain untouched by the system.
- While a Rollback is in progress, a user cannot trigger Migration, Deployment, or additional Rollbacks.

Data Tables/File Manager

- Individual Data Tables or File Manager files are affected in Rollback only if a version is contained within any snapshot between current state and the state being rolled back to.
    - An individual Data Table will only be included in the snapshot taken immediately prior to a migration if it has been selected for that specific migration. Therefore, this Data Table may be contained within multiple snapshots, only one snapshot, or no snapshot at all. The same follows for an individual File Manager file.
        - A Data Table will be rolled back if it is contained within a snapshot (i.e. has been migrated) between present state and the selected snapshot. If a Data Table is contained within multiple snapshots, it will be rolled back to the version stored within the oldest snapshot in this timespan.
        - A File Manager file will be rolled back if it is contained within a snapshot (i.e. has been migrated) between present state and the selected Rollsnapshot. If a File Manager file is contained within multiple snapshots, it will be rolled back to the version stored within the oldest snapshot in this timespan.
- Any Data Table or File Manager file not contained within a snapshot between current state and the snapshot being rolled back to will not be deleted nor altered by a Rollback.

Data Columns

- Rollback will remove a Data Column(s) if it is not contained within the snapshot selected for Rollback.
    - If a Data Column is removed via Rollback and was referenced in the Process Manager at the time Rollback is initiated, it will be removed from the Transaction Manager columns.
    - If a Data Column is removed via Rollback and was referenced in the Process Manager at the time Rollback is initiated, it will be removed from any custom Transaction Manager Views.
        - Affected views will be highlighted red in the drop-down menu.
    - If a Data Column is removed via Rollback and was referenced in a Report prior to Rollback, the Report will become invalid.
        - The invalid Report name will be highlighted red in the Report Manager.
        - The creator must manually remove deleted Data Columns used in the Report before the Report can be saved, scheduled, or run.
            - "[Attribute removed]", highlighted in red, appears where a reference to a deleted Data Column was used.
        - The system will send an email to the creator of any report that was made invalid by removed Data Columns during the Rollback process.
            - Scheduled Report emails will notify the recipient that errors have occurred and they must contact their admin.

General Enhancements:

- Error messages now clearly identify the location of the error when BML compilation fails during migration.
- New dialog window attached to migration button so the user may enter a migration description which is then displayed on the Migration Status page for reference.
    - Note: This field can be used for tracking granular details of a migration that can be useful for Rollback considerations.
- Data Columns can be deleted through Migration.
    - Deletion of a Data Column via Migration will have the same effect on Transaction Manager Views and Reports on the target site as if it were deleted in a Rollback (see above description).

NOTES

- Like standard migrations, Rollbacks will not affect transactions.
- Users will experience an increase in total Migration time due to the addition of the snapshot that must be taken to facilitate the Rollback feature.
- All snapshots are permanently deleted upon upgrade of a major Release. Snapshots are not deleted upon upgrade of an Update (dot Release) and thus can still be rolled back to.
- Only users with access to perform Migrations can also perform Rollbacks.
    - This includes Superuser and FullAccess users with Superuser permissions.
- Because Rollback performs similarly to Migration, when dependencies are removed for the migration and then a Rollback occurs, the results are equivalent to migrating without those dependencies.
- Configuration Backup functionality, located within the Configuration Deployment Centers, has been removed.
    - Existing backups will be discarded upon initial upgrade.
- Rollbacks will not fail if a simple table-based rule is reverted when using a table that no longer exists.
    - A warning message will be logged in the Rollback status.
    - Simple table-based rules will return null at run time. When the administrator next attempts to edit the rule, the application will request that the user select a new table.
- Snapshots can fail if data is missing or incorrect.
    - A Rollback will fail if a Commerce or Config attribute that is used in a data table validation is reverted.
        - An error message will state that the attribute does not exist. User must manually remove the validation from the target site.
    - Migration logs include errors produced when taking a snapshot.
- Status emails are sent to the Superuser after Rollback is complete.
- A snapshot is not saved if a Migration fails, and therefore the "Undo" button is not available.
- Migration records cannot be deleted from the Migration Status page if the "Undo" option is still available.

Known Behavior:

- When migrating, Util Libraries that are referenced by selected Commerce Processes and Config Families will be listed in a dependencies dialog pop up. An individual Util Library will be shown even if it is selected in the main window at the start of migration.
    - It is highly recommended that the user select the Util Library in the dialog pop up even if it has already been selected in the main window.

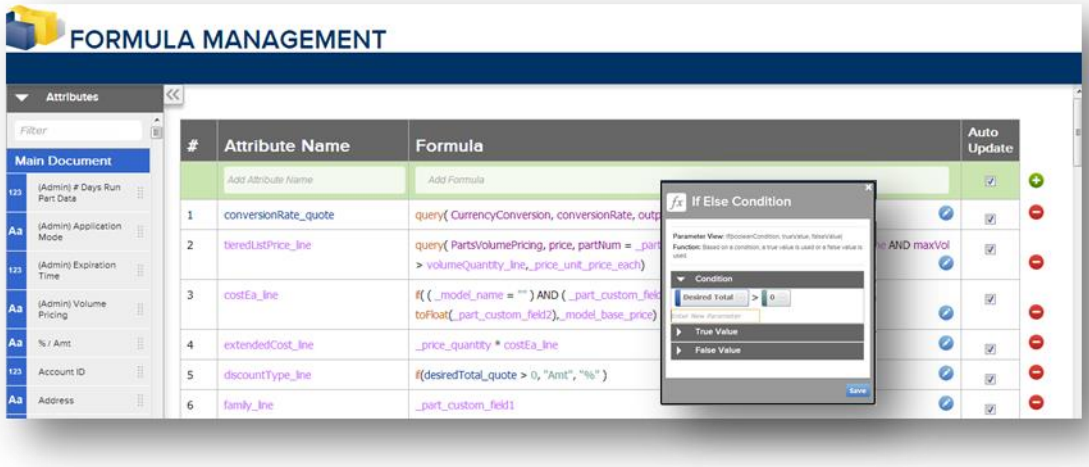
FORMULA MANAGEMENT

FORMULA MANAGEMENT ENHANCEMENTS

- Performance enhancements to improve response for quotes with a large number of line items and large sets of formulas.
- Select quote-level formulas to be executed upon adding line items to a quote.
- Line item default and auto-update quote function now have a pre and post modify, which aligns with 2013 Winter actions.
- Enhanced admin UI views within the Formula Management Center.
- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound).

SUMMARY

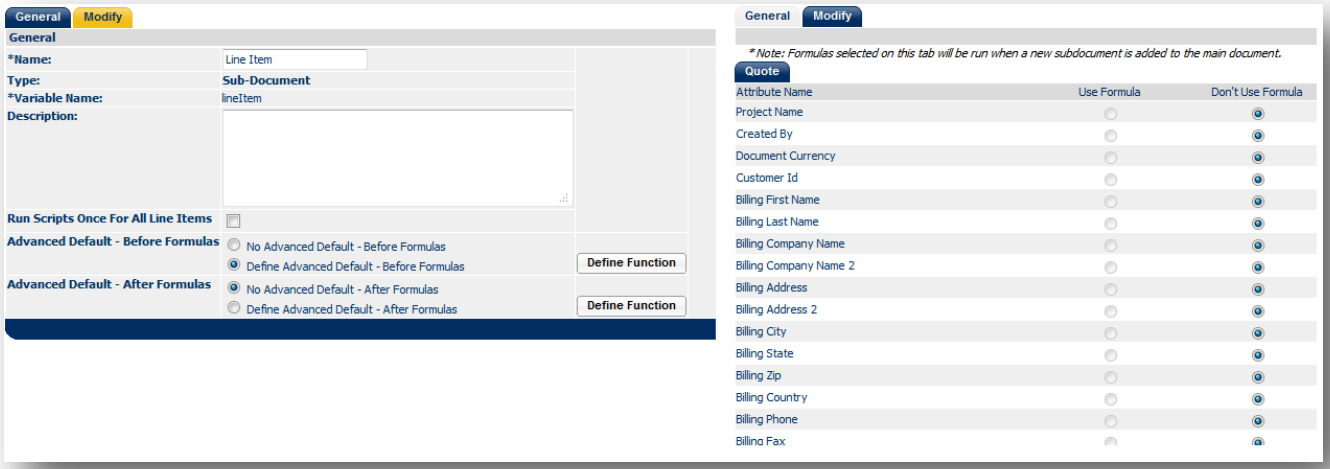
The following enhancements improve upon existing Formula Management. Complex user scenarios that previously required BML pricing calculations can now be done using Formula Management. These improvements enable administrators, proficient in Excel, to create complex pricing and discounting calculations without advanced rules and scripting.



ADMINISTRATION OVERVIEW

Formula Execution with New Line Items

- Quote level (main-document) formulas can be executed when users add a line item (sub-document) to the quote, regardless of whether the line item is a result of Configuration, Search, or Quick Key Line Items.
  - On the admin side, the line item Edit page now contains a Modify tab. This tab allows the admin to select which quote level formulas to be executed when a new line item is added to the quote.



- Navigation: Admin Home Page > Process Definition > Documents ("List") > Line Item link
- Formula execution occurs on the addition of a new line item.
  - When a new line item is added to a quote, the formulas that are defined for line item attributes will run to determine the default values of the attributes.

- Formulas defined for an attribute can be self-referencing.
  - Self-referencing formulas only run once.
- Line item default execution path now includes Formulas Execution and the ability for administrators to run Advanced Default- Before Formulas and Advanced Default- After Formulas.
  - Sub-document General tab includes a second Advanced Default script, named "Advanced Default- After Formulas."
    - The function previously known as Advanced Default has been renamed, "Advanced Default- Before Formulas."
  - Order of execution: Modify tab, Advanced Default- Before Formulas, Formulas, and Advanced Default- After Formulas.
- A new Modify tab is located within the line item Admin Detail page to allow the admin to enable a formula to execute per quote level attribute.
- The Modify tab on the line item page contains only quote attribute formulas.
  - Only quote level attributes exist on the line level Modify tab. For each of the quote level attribute the user can specify whether to use or not use formulas during a new line item addition.

Auto-Update: Flexibility with BML and Formula Management

- "Auto Update Before Formulas" option added to Quote Document admin page.
  - Allows functions to run before and after Formulas when the Commerce page performs an Auto-Update.
  - Navigate from: Admin Home Page > Documents ("List") > Main-document link.
- Previously existing Auto-Update has been changed to post-formula Auto-Update function.

Formulas in Reconfiguration Scenarios

- Reconfigure action is split into one parent (Reconfigure) and one sub-action (Reconfigure Inbound). Previously, there was only a single Reconfigure action that required conditionals to segregate logic.
  - The parent action contains all things common to both Reconfigure actions and its own identifiers: Label, Variable Name, Description, Action Icon, Show Loading Dialog, Layout Path, and Document Views Tab.
  - The Reconfigure Inbound action only contains what is specific to it.
    - Upgrade to 2013 Autumn will preserve existing implementation behaviors. As a part of the upgrade, the pre-existing Reconfigure action BML scripts will be moved to the new Reconfigure action.
  - This allows administrators to execute a formula per attribute on:
    - Outbound paths from a Commerce Transaction.

        - Formula will run immediately after user clicks Reconfigure action in Commerce and before user enters Configuration.
    - Inbound paths to a Commerce Transaction.
        - Formula runs after user clicks Save in the Reconfiguration and before being returned to the Commerce page.
- Formula execution is consistent with the existing order of operations for inbound and outbound executions.

NOTES

- Unnecessary validation of inputs into formulas has been removed.
- Admins can now view formulas in a Label or Variable Name Display Mode by clicking a new button in the Formula Management Administration page.
- Upon upgrade to 2013 Autumn:
    - Legacy sub-document Advanced Default is replaced by Advanced Default- Before Formulas.
    - Legacy main-document Auto Update becomes Auto Update After Formulas.
    - This ensures that previous execution order is preserved.
    - Because new Reconfigure action IDs may be generated, if anything is calling a Reconfigure action using its ID, the reference may have to change.

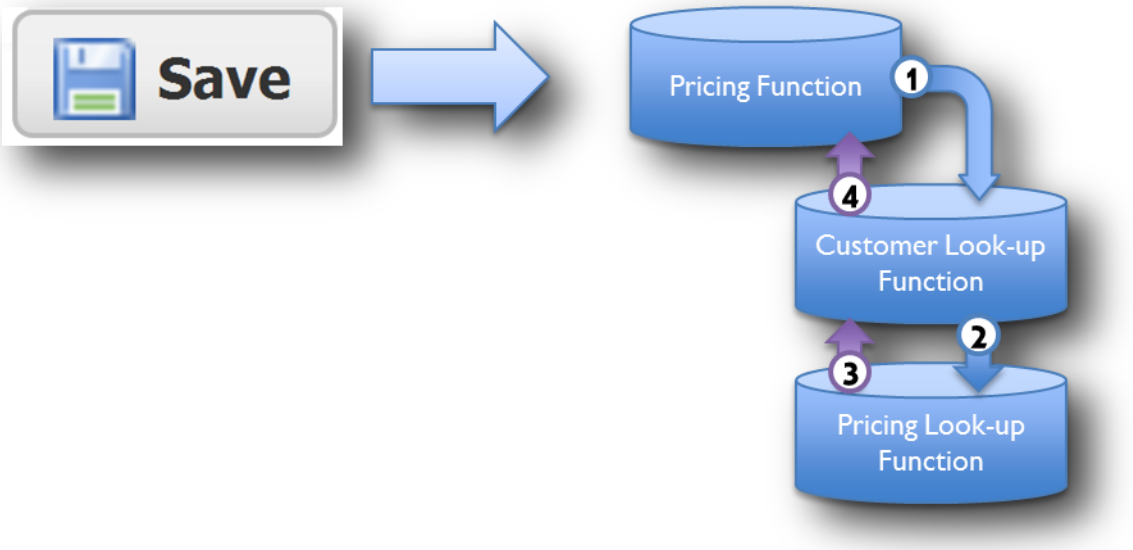
FUNCTIONS

FUNCTION TO FUNCTION CALLS

- Commerce Library functions can call other Commerce Library functions.
- Util Library functions can call other Util Library functions.

SUMMARY

Function to Function calls allow administrators to compartmentalize BML when dealing with complicated configuration or quoting scenarios. This feature will assist with the organization of BML and provides a solution to the compiled Java class size-limit issue. Function to function calls mimic the behavior of a Modify function calling a Library function.



ADMINISTRATION OVERVIEW

- Library Function(s) tab appears on Util and Commerce Library Function Editor.
  - Contains list of Commerce and Util functions, Preview, and ability to insert function into editor.
    - Commerce Libraries will only be listed in the Commerce Library Functions.
      - Util Libraries will be listed in both.
  - Choose drop-down menu option to import Util and Commerce Library Functions, excluding the function the user is currently in.
- Commerce Library functions can call upon a separate Util or Commerce Library functions. Util Library functions can call other Util Library functions.
  - Util and Commerce Library functions cannot self-reference.
    - Recursive calling of the same Util and Commerce Library functions will fail and result in a compilation error when called at any point in the reference chain. Recursive calls of dependency cycles are not allowed.
    - Util and Commerce Library functions will not appear in the Import list for themselves.
  - Recursive validation is performed during the following for Util and Commerce Library Functions: Validate, adding, applying, or updating a function.
  - The Util and Commerce Library Editors include tabs to select other Util and Commerce Library functions, when applicable, for import.
  - Only Util Library functions will appear in the Util Library Editor.

- Commerce Library functions and Util Library functions will appear in separate folders in the selection list within the Commerce Library Editor.

- Function to function link appears on the Related Rules page.
  - When it is referenced by other Utils it will be displayed on the Related Rules page.
- Util-to-Util and Commerce-to-Commerce calls will mimic the existing call format for any Library function.
  - General structure: util.functionName() and commerce.functionName().
    - Parameters will be passed within the parenthesis.

MOBILE

BigMachines browser-based solution for Mobile offers customers: better security control, lower maintenance cost, and improved user experience. Mobile users can present complete quotes to a new prospect, configure quotes, and create/edit quotes on a pre-configured or newly configured item. Mobile will benefit the sales users, who are out in the field, the most. It improves win rate, reduces sales cycle time, and involves the end customer in a collaborative quoting experience. Organizations that leverage collaborative selling on tablets also reduce their sales expenses and differentiate themselves among their competition.

- Mobile is enabled for all customers by default. Licensing considerations may be made in future versions.
- Mobile functionality includes: Punch-Ins, Pending Configurations, and Parts Search. Shopping Cart and Return To Opportunity have been excluded.
- Parts Search and Parts Search Results have a Mobile optimized page.
- For optimum performance, it is recommended that Landscape Orientation is used.
- User is automatically redirected to Desktop Version of page when Mobile Version does not exist or user does not have a Mobile license.
- Mobile Navigation button, which appears in the top left corner, contains links from Commerce and Subheader sections, depending on the user type. The admin can set permissions here: Navigation Menus > List Links > Customize Menus.



MOBILE CONFIGURATION

- Mobile layout support.
- Utilizes device specific field display.
- Optimized page load and design.

SUMMARY

With the introduction of BigMachines Mobile, layouts are automatically configured to recognize that user is on a mobile device.

ADMINISTRATION OVERVIEW

User Side:

- User can scroll through lists using vertical swiping and perform horizontal swiping on tabs and array attributes.
  - The arrays have only one array-type vertical scaling option whether the array is tabbed or horizontal.
  - Array Control Attributes must be located on the same tab as the Array Set they control.
- On-demand tab loading: only the content of tab that is being viewed is loaded.
  - Tabs are located in the top panel.
  - If there are more than five tabs on the page, only the first five are displayed; the user can swipe to see the other tabs or use the integrated list menu.
  - A button at the bottom of each tab allows users to navigate to the next tab.
- Array pagination: 10 array indices per page.
  - Rules will run on all indices, each index that contains a message will be highlighted and only the selected index messages will be displayed to eliminate vertical scrolling.
- Layout is organized into a one column to reduce the need for horizontal scrolling.
- Recommended Items and Pricing Sections appear in the collapsible sidebar on the right hand side.
- Mobile functionality takes advantage of the native functionality of a tablet device. Examples: Date Picker and drop-down menus.
- Action bar at bottom of the tablet page contains the following buttons, when applicable: Next, Previous, Start Over, Save, Cancel, Add to Quote, Return to Quote, Create Quote, and Add to Favorites.
  - Mimics existing Desktop functionality.
  - Contains Pipeline Viewer for FullAccess user types.
  - Action bar displays up to five buttons.
    -  "More" button contains any additional buttons.
  - Scenarios in which the action bar is used:
    - Next and Previous: used in multi-node Config flows.
    - Save and Cancel: used in Reconfiguration contexts.
    - Create Quote: only used when a Commerce transaction has not been created prior to Config.
      - This action is temporarily hidden until "Pending Deletes" is undone or an update to "Delete" is performed.
    - Update: saves current values and runs rules again with the current Config values.
      - Any additional Invocation buttons will be added to the action bar.
- All Configuration rules will fire on Mobile as they would on Desktop.
  - Any attribute that is on the Desktop layout, but not on mobile layout behaves as if it is hidden on Mobile.
    - All attributes that are on the Desktop layout will be included in the Pipeline Viewer.
    - If a constraint rule fires on a "hidden" attributes, an internal constraint error will occur.
- When AJAX-enabled attributes are changed, AJAX rules run and the results display as expected.
  - Pricing attributes will update the pricing information.

Admin Side:

- Mobile layouts are created within the Flow Properties page.
  - Mobile layouts are initially generated to contain the contents of the Desktop layout.
    - The administrator may then edit Mobile layouts to contain fewer attributes than that of the Desktop layout. However, the Mobile layout cannot contain attributes not on the Desktop layout.
  - In order to regenerate the Mobile layout, the admin must delete the Mobile layout and restart the "Create Mobile Layout" process.
- Array sets can be dragged-and-dropped from the sidebar into groups.
  - Array sets contain a Delete button, but no Edit button.
  - Attributes within the sets contain Edit buttons, but no Delete buttons.
    - Array sets are ordered according to the associated set order.
- Multiple column layouts on Desktop will be converted into a single column on Mobile.
- The admin Mobile layout will be auto-generated from the Desktop layout the first time it is created.
  - Once a Mobile layout is created it becomes a separate layout that must be updated independently in order for user side changes to take effect.
    - The only attributes available to a Mobile layout will be the attributes which are already present on the desktop layout.
  - When an attribute exists on Desktop, but not on Mobile, the attribute is still present, just hidden, on the mobile layout.
    - Any rules which are related to that attribute will still fire when users access the Mobile layout.
    - If a required field is not added to the mobile layout and it lacks a default value, the user will be unable to create a quote.
- Attributes can be moved, edited, and deleted on the Mobile layout.
- Layout Properties has been moved from the Config Flow Editor page to the Desktop and Mobile Layout Page.
- "Mobile Device Enabled" system attribute is now available as a rule input for Recommendations, Constraint, and Hiding rules.
  - Attribute will return "True" if the following conditions have been satisfied:
    - User is on Mobile device.
    - Administrator has enabled Mobile access for user.
    - Administrator has defined Mobile layout.
  - System attribute returns "False" if one or more of the above conditions have not been met.
- An alternate stylesheet can be uploaded for Mobile.
  - All of Mobile utilizes one alternate stylesheet.
- Action bar at bottom of tablet contains the following buttons: Remove Layout, Layout Properties, and Preview.

NOTES

- Although mobile layout is visible in iFrame, optimal user experience is achieved through the full window.
  - The "Occupy entire window when the site is inside a frame" parameter controls the iFrame setting for both Desktop and Mobile, located in General Site Options.
    - To control this parameter user must select "True" to occupy the entire window when the site is inside a frame.
- Migrations will fail if the destination site and source site's Mobile Enabled settings are not identical.
- Pages with a large number of attributes or array rows could see performance issues.
  - It is highly recommended that users break up large sets of data to mitigate potential performance issues.
- Configuration calculators are not supported on Mobile.

Custom JavaScript Revisions:

- Sites using any custom JavaScript must revisit the areas in which custom JavaScript has been implemented in order for Mobile layouts to appear as expected. Custom JS is usually implemented in the following areas:
  - Headers
  - Footers
  - JS files uploaded to the File Manager through Config or Commerce HTML Attributes

Known Behavior:

When a multi-select menu attribute with the Display Type, List, has a blank value it cannot be unselected. If the blank value is constrained on a non-required attribute, the user cannot continue. This only affects Mobile. This issue can be avoided by implementing the following:

- Make the attribute required.
- Choose a different display type:
  1. Navigate from: Admin > Product Family > Configurable Attributes Administration List > select Name (Variable Name) > General tab
  2. Select either "Checkbox Horizontal" or "Checkbox Vertical" from the **Display Type** drop-down menu.

MOBILE HOME PAGE

- Mobile Home Page mimics desktop Home Page functionality.

SUMMARY

Mobile Home Page mimics the functionality of the desktop Home Page, but utilizes finger interaction, "gestures", instead of typical desktop mouse functionality. Administrators can set permissions for the Mobile Home Page.

ADMINISTRATION OVERVIEW

User Side:

- Users can swipe horizontally through Product Family tabs.
- Pending Configuration has been optimized for Mobile display; they only appear when user has Pending Configs.
  - The Pending Configs page can be accessed by swiping from the Model Punch-In page or on the Navigation link at the bottom.
- Parts Search and Parts Search Results has been optimized for Mobile. Part Search performs a simple search of the user-entered String against Part Descriptions and custom fields.
  - Part Search displays the first 25 results. Click the "More" icon to see the next 25 results.
    - Parts selected from Parts search will continue to be selected/accumulate, even if another search is performed, until user returns to the quote or Commerce.

Admin Side:

- "Mobile Device Enabled" system attribute is now available as a rule input for Home Page permissions.
  - Attribute will return "True" if the following conditions have been satisfied:
    - User is on Mobile device.
    - Administrator has enabled Mobile access for user.
  - Will return "False" if any of the above conditions have not been met.
- Model Punch-In descriptions are pulled from the text field in the Model Properties page.
- Mobile Home Page display includes Model Punch-In selections with labels, descriptions, and images.
  - Images can be added to the Mobile Punch-In using the image Browse/Upload function, located on the Admin Home Page.
    - Recommended maximum image width and height: 235px by 150px.

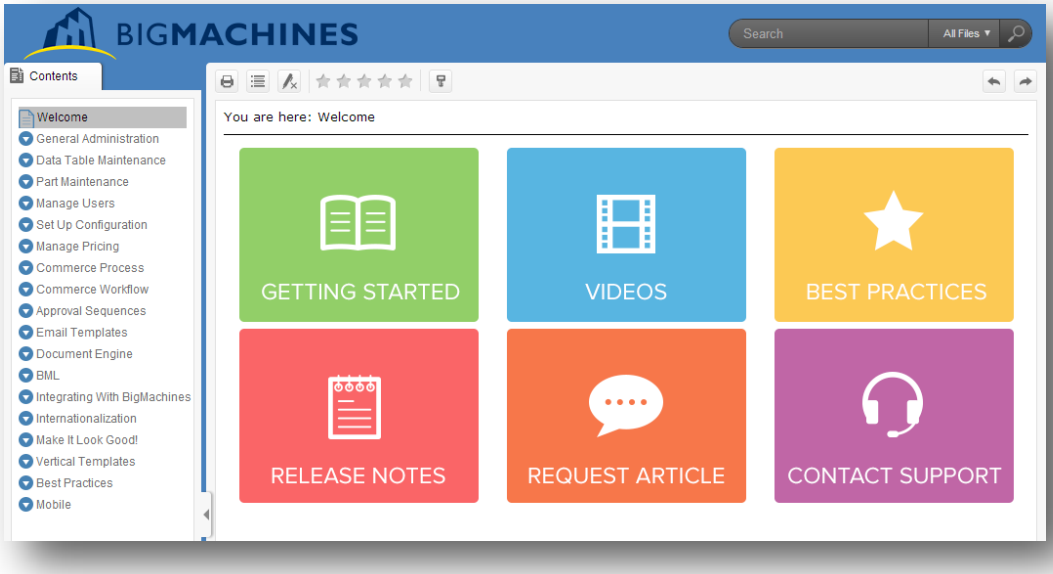
HELP SYSTEM

HELP SITE

- Improved usability: new Search functionality, updated navigation, article reorganization, and expanding togglers.
- Documentation Team interaction: topic rating, comment boxes, Customer Article Submissions, and "Contact Us" direct email link.

SUMMARY

A new Help site featuring improved usability and interactivity has been implemented using MadCap Flare. The BigMachines Help site is accessed via SSO by clicking the "Help" link in the upper right hand corner of the BigMachines site.



ADMINISTRATION OVERVIEW

- **Easy Navigation:** Article reduction has been facilitated by the use of expanding togglers.
- **Topic Reorganization:** Articles are grouped according to your end goal for improved findability.
- **Feedback & Topic Rating:** Interact with the Documentation Team by submitting comments and rating articles.
- **Customer Article Submissions:** Submit articles that you would like to see included in the Help.
- **Release Notes:** PDFs of Release Notes are available.
- **Customer Support Access:** Direct link to Customer Support is available on the Welcome page and at the bottom of every article.
- **Search Filters:** Search filters narrow down results according to the following categories:
  - Advanced Use Case
  - Concepts
  - Customer Articles
  - FAQ
  - How Do I...?
  - Troubleshooting
- **Synonyms:** Dictionaries including synonyms of frequently failed search words have been created to make the Help more intuitive.

GENERAL ENHANCEMENTS

SECURE FILE MANAGER

- Enhanced capability to set File Manager folders as secure.
- No upgrade impact; existing functionality will remain.
- Improved migration capabilities for File Manager references (Doc Engine, BML).

SUMMARY

BigMachines has addressed past vulnerabilities and improved upon existing File Manager functionality by providing administrators with the ability to secure files at the folder level within the File Manager. This ensures that secured files may only be accessed when logged into a BigMachines session. A checkbox has been added to designate a folder as secure.

ADMINISTRATION OVERVIEW

- Select the checkbox and click Save to designate a folder as secure.
- The folder is automatically updated with the values and security setting that the administrator has chosen in the folder. This selects the security setting.
  - The administrator can designate any folder they have added as secure.
- Secured files will contain a $SECURE_PATH$ variable.
  - The file path will be used instead of the file ID to locate secure files.
  - When the Image Server is called with both a file path and a file ID, it will:
    - If the file has a file path, the file will be returned regardless of the file ID.
    - If the provided file path cannot locate the file, the file will be located via the file ID.
    - If the file cannot be found via file path nor file ID, no image will be returned and a broken image will be displayed.
- When migrating, the security settings from the source folder will be transferred to the target folder.
- Once a folder is secured, the following icon will replace the checkbox alongside the image: 
- A new DiffType indicates if the security settings on the target and source site don't match a particular file or folder.
  - Checkboxes will be disabled for files whose folder security does not match that of the high-level Diff Tree View in Migration.
    - The text "Security does not match the source" will be displayed next to the folder whose security settings do not match.

NOTES

- The Search functionality of the file manager has been unchanged. It returns secured and unsecured folders alike.
- The Doc Engine File Manager dialogs will display images and return generically reference-able links or server links.
- A confirmation dialog will appear when an administrator is updating the security settings.
  - Toggling the security of a folder is potentially costly due to file referencing, which can result in broken links.
- When performing a Bulk Download of the File Manager, both secured and unsecured files will be downloaded.
- When performing a Bulk Upload of the File Manager, the user cannot specify files to be uploaded to a secure folder.
  - If uploading to an existing folder, the pre-existing security settings will remain.
  - If uploading to a new folder, a folder will be unsecured by default.
- The user cannot perform a bulk change of a folder's security settings.
- Legacy paths will continue to perform as expected.

- The inner server paths will continue to perform as expected; if the user clicks on the Secure path, the IDs will still work no matter if the user switches bath and forth between secure folders.
- If site is upgraded and user changes a folder to Secure, the path must also be changed.

Expected Behavior:

- When updating a folder's security settings, the "Date Modified" of all the files located within the folder will be updated to the current date.

GENERAL ENHANCEMENTS

LINE ITEM PERFORMANCE ENHANCEMENTS

- Focus on tuning performance for up to 2000 line items.
- Optimized memory handling at both client and server level for large quotes.
- Tune application code when loading line item information for a quote.

SUMMARY

BigMachines has addressed the needs of customers that sell larger opportunities with a larger volume of products; specifically between 500-2000 products. Enhancements included also address the large volume of products introduced during renewal quotes. The BigMachines application can now handle a large amount of line items with a quicker response. CRM Integration with a large number of line items has been optimized.

ADMINISTRATION OVERVIEW

- Server response time was benchmarked using the BigMachines Base Template site.
  - Performance has been improved to target 1/9th the execution time to process approximately 2000 line items on any quote actions, including: quote creation, quote saving, and quote price application.
    - Please note that depending on specific implementation and browser speeds of customer sites, the results may vary.

GENERAL ENHANCEMENTS

ORACLE FUSION

- Standard authentication mechanism using Fusion-defined SAML approach.
- Support different end points (URLs) for integration with different CRM objects (available at Integration XSL level).
- Global logout implemented for all Oracle Fusion SSO functionality.

SUMMARY

BigMachines has added support for Oracle Fusion. This is in addition to already supporting: Salesforce.com, Oracle On Demand, and Microsoft CRM. Users will be authenticated from BigMachines to Oracle Fusion via SSO. System architecture has been updated to support calls from multiple endpoints.

ADMINISTRATION OVERVIEW

- XSL endpoints are defined per integration XSL.
  - The user goes to the Edit Integration XSL page where the SOAP call may be specified.
- Fusion package configuration will require additional Oracle documentation during setup.
  - Like the Oracle On Demand integration, some of the setup steps on Fusion are manual.
- SAML SSO authentication added.
  - Setup follows the standard Federated Authentication setup found in the Help.
    - An Oracle Fusion setup requires the addition of a RelayState URL Punch-In for the relevant quote following the standard Federated Authentication setup.
  - Global logout is available and can be initiated from BigMachines and Fusion.
  - Setup tested with Oracle Federation; Oracle's SAML server.
- Multiple endpoint capabilities:
  - A textbox to list the URL endpoint for each XSL.
    - Outbound URL for XSLs validated against a BigMachines-maintained whitelist.

NOTES

- Proxy logged-in admins will be returned to the Users list when click Proxy Logout is clicked.
  - This occurs regardless of SSO logout URL specifications.

General Enhancements

Translations Bundle

- French
- German
- Spanish

Summary

BigMachines, working with a professional translation company, has translated site content into three languages: French, German, and Spanish. All product templates and pre-existing BigMachines content users receive at implementation is available in translation.

Administration Overview

- The addition of the translations bundle does not affect current customer functionality.
- User is not required to make any selections, once site language preference established, customer site will appear with translations in place.

Additional Enhancements

Login allows periods

Users can now use periods in their BigMachines usernames.

Price Book columns increased to accommodate more currencies

BigMachines has increased the number of supported currencies within the Standard Price Book by increasing the column size of the Price Book Entry ID. Previously, the Entry ID field was limited to 2500 characters, which only allowed for 115 currencies to be supported. Now, Partner Organization Standard Price Books can accommodate as many currencies as are supported on a BigMachines site.

Mass Update function can be performed on a subset of steps

Now, when performing a Mass Update, users can filter which transactions the update applies to. Users can select the steps to which the Mass Update will be applied. Only transactions on that specific step(s) will be updated.

BMQL: return Part Description in specified language translation

Administrators can now override the user's language with a specific language to retrieve a specific translation of the Part Description using BMQL. Language parameters can be specified using a local variable or attribute variable. If a language value is set, that value will be returned; if the value is blank, the site's base language will be called. If the BMQL query doesn't contain a language parameter, the user's session language will be used.

Reconfigure in a different language

Users can reconfigure a model in a language other than the quote language, which is determined by the quote creator's language preferences. Configuration Rules will utilize the language of the user performing the Reconfigure; Configuration Flow Rules will utilize the original specified language. Rules will only behave differently if the Configuration Rule inputs depend on the system language Attribute.

BML function keys() added

The BML function, keys(), has been added to the BML Editor. This allows administrators to retrieve an unordered String Array of all keys found within a Dictionary. If the Dictionary does not contain any keys, an empty array is returned. Refer to the BML Function Wizard for syntax examples.

"Web Services Only" user permissions checkbox

A "Web Services Only" checkbox has been added to the Permissions section of the User Administration page. Only FullAccess users with the ability to create/modify users can change this setting. When the "Web Services Only" checkbox is selected for an internal user, that user may only make Web Services calls to the BigMachines site; logging in through the web interface will not be permitted.

EXPORT ATTACHMENT ACTION LANGUAGE POP-UP

Export Attachment-type actions have been added to the "Show Select Language Preference menu for Print and Email actions" Commerce setting. Once an attachment is exported in Commerce, a popup dialog will appear allowing the user to choose a system-selected language to export to the CRM. In order to take advantage of this feature, the document must use the system-selected language attribute; the selected language defaults to the user-preferred language. Note: like other actions, the document will not be automatically translated.

ENHANCED SECURITY

Security has been increased for the exportation of Commerce Transactions and other data.

THIRD PARTY SOFTWARE

Copyright information and related attribution files have been updated for 2013 Autumn. For more information regarding third party software copyrights, consult the BigMachines Help.

RESOLVED ISSUES

USERS/GROUPS/SECURITY

Known Behavior:

- If a site exceeds the license limitations on any of the license types (Internal User & Partner Organization User), the administrator will not be allowed to create any new users until the license count is decreased and no longer exceeds the license limit.
- When the license limit is reached, the administrator will receive an error message instructing him to contact Customer support and purchase additional licenses.

mod_security allows Previews of Home Page header and footer (SFDC Bug ID: 22337)
In Version 11.1, when a user attempted to Preview a Home Page header and footer, mod_security prevented access to this page and a 400 error appeared. Now, mod_security will be disabled and no errors will occur during a Home Page header and footer Preview.

Partner Company name change will be reflected in each user's profile (SFDC Bug ID: 29990)
In Version 12.1, if a Partner Company name was modified, the name change would not be automatically reflected in each user's profile. Now, a Partner Company name change will be automatically populated into each user's profile.

Admins proxy-logged in as users will be redirected to Users list after proxy logging out (SFDC Bug ID: 44365)
In 2013 Spring, when an administrator, who was proxy-logged in as a user, clicked Proxy Logout, they were logged out of their BigMachines instance. Now, when an administrator, who is proxy-logged in as a user, clicks Proxy Logout, they will be redirected back to the Users list.

Delegated approver username can be greater than 25 characters (SFDC Bug ID: 45870)
In 2013 Winter Update 3, the delegated approver username could not be greater than 25 characters. The combination of the First Name and Last Name fields was irrelevant. This occurred because SQL was not run during a build. A site refresh has been performed and now the delegated approver username can be greater than 25 characters.

Duplicate Partner Organizations will not be created when "Add" is double-clicked (SFDC Bug ID: 47820)
In 2013 Spring, when a user double-clicked the "Add" button on the Partner Organization page, a duplicate Partner Organization was added. Now, when a user double-clicks the "Add" button on the Partner Organization page, a duplicate Partner Organization will not be added.

Partner passwords can be copied into password field using right click function (SFDC Bug ID: 48599)
In 2013 Spring, partner passwords could not be pasted into the password field using the right click function. Now, partner passwords can be pasted into the password field using the right click function.

Updates to supported language property will take effect without requiring a site restart (SFDC Bug ID: 49194)
In 2013 Spring, when the supported language properties were updated, a site restart was required for the updates to take effect. Now, updates to the supported language property will take effect without requiring a site restart.

COMMERCE

Known Behavior:

- Across Config and Commerce, zero and blank integer values are treated as separate values.
  - Zero = 0
  - Blank = “”

Functions without line items on the quote can be used for debugging (SFDC Bug ID: 17442)
In previous versions, when a user debugged a function, an error message appeared because there were no line items on the quote that was being used for debugging. Now, users will not receive error messages if there are no line items on a quote that is being used.

Floating navigation bar will not affect page height (SFDC Bug ID: 17731)
In Version 11, when a user scrolled vertically on a page with a floating navigation bar, the page would change height, causing the docking of the floating navigation bar and the appearance of the vertical scroll bars to fluctuate and the pixels at the bottom of the screen to disappear. Now, the floating navigation bar will not change the height of the page.

Back button in Import/Export Commerce Process returns users to previous page (SFDC Bug ID: 17943)
In Version 11, when a user clicked the Back button within Import and Export Commerce Process pages, they would be returned to the Admin Home Page. Now, clicking the Back button will return the user to the previous page.

Sticky action bars will perform as expected (SFDC Bug ID: 20654)
In Version 11, when the sticky action scrollbars were moved from relative to fixed status, the page height would resize, causing a vibration effect. The fix in SFDC Bug #17731 has resolved this issue. Now, the vibration effect will not occur.

Accessing Quote Number via Salesforce on an iPad will perform as expected (SFDC Bug ID: 20693)
In Version 11.04, when a user opened a quote via Salesforce using an iPad and clicked on a Quote Number, an error icon would appear. The error only occurred when a user logged in via Salesforce. This issue has been resolved.

Company name will appear in the XML for the Transaction History (SFDC Bug ID: 21227)
In Version 11, the name of the user company would not appear in the Transaction History. Now, the company name will appear in the XML for the Transaction History.

Commerce Process attributes using “Elements” as varname will invoke properly (SFDC Bug ID: 22807)
In Version 11.1, when a user utilized “Elements” as a varname, the Commerce Process could not be invoked until “Update” was clicked. Now, the following error message would appear, “Elements is not a valid Variable Name. Elements is a reserved keyword.”

“Search Again” button available on non-full access user Account Browse pages (SFDC Bug ID: 23113)
In Version 11.1, the “Search Again” button is missing from the Account Browse page for non-full access users. Now, the “Search Again” button is available on non-full access user Account Browse pages.

Commerce attribute Help Text changes will only appear after deployment (SFDC Bug ID: 23305)
In Version 12.1.0, when a user altered the Help Text of a Commerce attribute without deploying, the change would be seen on the user-side immediately. Now, changes in the Help Text will only appear once the values have been deployed.

Clicking “Add to Transaction” multiple times will only add a part once (SFDC Bug ID: 26066)
In Version 11.1, when a user clicked the “Add to Transaction” button multiple times, multiple line items would be added to the quote. Now, if a user clicks “Add to Transaction” multiple times, the line item will only be added once.

“List” will display Commerce attribute Related Rules (SFDC Bug ID: 28245)
In Version 12, when a user attempted to “List” the Related Rules of a Commerce attribute, a null ID error message would appear. Now, users can “List” Commerce attribute Related Rules.

Illegal file uploads will not lock quotes (SFDC Bug ID: 28306)
In Version 12, when a user uploaded an illegal file, no actions could be triggered (e.g. Delete) and the quote became locked. This issue has been resolved.

Commerce Library Debugger Import button performs as expected (SFDC Bug ID: 31471)
In Version 12.1, when a user clicked the Import button a second time in the Commerce Library Debugger, the last line item entry copied the first line item's values. This issue has been resolved.

Last paginated page will display correctly when page size is adjusted (SFDC Bug ID: 31857)
In Version 12.1, when a user adjusted the page size on a quote, the last page displayed incorrectly; the line items would not be distributed evenly across the paginated pages. Now, when a user adjusts the page size, the last page will display correctly.

Clicking "Email PDF" will render sent PDF properly (SFDC Bug ID: 32425)
In Version 12.1.0, when a customer emailed a quote with an attachment using an email system, the PDF would not render properly when the recipient attempted to open it. This occurred because BigMachines had set the content type as Plain Text; most email systems recognized when an attachment was being emailed, but some did not (e.g. Zimbra). Now, when users click "Email PDF" using an email system, the PDF will render properly.

Commerce Processes with email templates can be cloned (SFDC Bug ID: 35578)
In Version 12.2, users could not clone a Commerce Process if any email templates existed. Now, users will be able to clone a Commerce Process if an email template exists.

Approval History will show "Revised the Quote" when user clicks "Revise" after "Submit for Approval" (SFDC Bug ID: 38287)
In 2013 Winter, when a user clicked "Revise" to edit a quote after clicking "Submit for Approval", but before approving or rejecting, the quote would appear to be "Submitted for Approval" in the Approval History. Now, when a user click "Revise" to edit a quote after clicking "submit for Approval", but before approving or rejecting, the quote will appear as "Revised the Quote" in the Approval History.

Trailing zeroes will conform to float variable decimal precision (SFDC Bug ID: 38712)
In previous versions, float variable decimal precision behaved inconsistently regarding trailing zeroes. For example, when a user would set the float variable decimal precision to four places in the Attribute General Properties, a value of "12.25" would be displayed as "12.25" on the user side. Now, float variables will gain the correct number of trailing zeroes up to the decimal precision set in the Commerce variable properties; for example, when the user sets the float variable decimal precision to four places, a value of "12.25" will be displayed as "12.2500" on the user side.

Reconfigured Price Books using Pricing Rules instead of Rec Item Rules retain set values (SFDC Bug ID: 39207)
In Version 12.2.2, when a user reconfigured a Price Book using Pricing Rules instead of Recommended Item Rules and then clicked "Revise Quote", the Price Book would fail to retain the values it was set to. Now, reconfigured Price Books using Pricing Rules in place of Recommended Item Rules will retain set values.

Formulas can be used to output set attributes (SFDC Bug ID: 39974)
In 2013 Winter, address set fields could be used as formula inputs on main quotes, but when a user attempted to output an attribute within the set, the autocomplete appended "_quote" creating the wrong output formula and resulting in an error message. Now, users can use formulas to output set attributes.

Line item constraints accounted for when main-document evaluates rules (SFDC Bug ID: 40867)
In 2013 Winter, if a line constraint rule was added to a Commerce Process, the constraints would appear incorrectly on the main-document actions. The main-document would evaluate the rules for the document, but not the rules for the line items. Now, line item constraints will be accounted for when the main-document evaluates the document rules.

Skip Approvers and Approval Reason can be reset (SFDC Bug ID: 42381)
In 2013 Winter Update 3, when a user made an approval with a condition, that approval would not be reset when transactions were revised. This issue has been resolved. Now, an approver's approval will be reset when a transaction is revised.

Changing input Attribute will trigger Auto-Update (SFDC Bug ID: 42681)
In Version 12.2, when a user changed a Hiding Rule's action Attribute, Auto-Update would be triggered. Now, when a user changes the Hiding Rule's action Attribute, Auto-Update will not be triggered; changing the input Attribute will trigger Auto-Update.

Cursor will move according the browsers default behavior when pressing Tab in Customer Contact (SFDC Bug ID: 42682)
In 2013 Winter Update 4, when a user was entering Customer Contact information and pressed the Tab key to enter information into the next field, the cursor would not move according the browsers default behavior. Now, the cursor will move according to browser default behavior.

When enabled the loading icon and grey screen appears when saving a reconfigure action (SFDC Bug ID: 43149)
In 2013 Winter Update 3, when a user saved a Reconfigured model, the loading screen would not appear even when enabled. Now, the loading screen will always appear when saving a Reconfigure action, when enabled.

Attribute will only be added once regardless of the number of times "Add" is clicked (SFDC Bug ID: 43155)
In several versions (Version 12.2, 2013 Winter, and 2013 Spring), when a user created a new Commerce Attribute and clicked "Add" several times, the attribute would be created multiple times using the same order number. Now, when a user creates a new Commerce Attribute and clicks "Add" several times the attribute will only be added once.

Defining Notifications on a Process Definition Step search filter will work as expected (SFDC Bug ID: 43880)
In 2013 Winter, when a user applied a search filter to Process Definition Steps and clicked "Define Notifications", a TypeError error message appeared. Now, when users click "Define Notifications" after applying a search filter to Process Definition Steps, no error message will appear.

Functions with new line item or quote attributes will be validated (SFDC Bug ID: 44382)
In 2013 Spring, when a user added additional attribute references to Commerce Library functions, the Data Table ID would be too large to be stored. This issue has been resolved by increasing the maximum column size.

Boolean attribute checkbox display type performs as expected (SFDC Bug ID: 44575)
In 2013 Spring, when a user deployed a Boolean attribute checkbox display type that defaulted to True (checkbox was selected) and then unselected the checkbox, the checkbox would remain selected upon submittal. Now, when a user deploys an unselected Boolean attribute checkbox, it will remain unselected upon submittal.

XML tag, <user_class> will not appear in Print and Integration XMLs (SFDC Bug ID: 45027)
In 2013 Spring, a new XML tag, <user_class>, appeared in the Print XML and Integration XML; this was sent on step transitions. This issue has been resolved; users must remove their current workarounds (XSD regeneration) before upgrading. Now, the XML tag, <user_class>, will not appear in the XMLs.

Superusers can add additional Commerce Transaction Views (SFDC Bug ID: 45672)
In Version 12.2.3, when a Superuser attempted to add an additional Commerce Transaction View, the following error message appeared, "Only the superuser can set a search as shared." Now, when a Superuser attempts to add an additional Commerce Transaction View, no error message will appear.

Unsaved Line items retained when validation is triggered (SFDC Bug ID: 45953)
In 2013 Spring, unsaved Line items added via Quick Key would be removed when a validation was triggered. Now, unsaved Line items are retained when a validation is triggered.

Subdoc attribute variable names beginning with "_" and followed by variable name will not be removed when importing a transaction into Commerce (SFDC Bug ID: 46959)
In 2013 Spring, when a user imported a transaction and the subdocument attribute variable name began with a "_" and was followed by the variable name of the subdocument (e.g. "_line_item_comment"), the subdocument variable name would be removed from the attribute variable name during the Commerce formula execution. Now, if a subdocument attribute variable name beginning with a "_" and followed by the variable name of the subdocument is imported, the system will check that the attribute is in the subdocument; if it is, nothing will be removed or prepended.

CONFIGURATION

Known Behavior:

- User will be prevented from creating or adding Quotes from Configuration to Commerce if mandatory items are missing.
  - This aligns with previous expected behavior.
  - In 2013 Spring, the user was able to create and add Quotes from Configuration to Commerce when mandatory items were missing.
- Across Config and Commerce, zero and blank integer values are treated as separate values.
  - Zero = 0
  - Blank = ""
- Attribute range calculations are now validated on the Value Punch-In. For example, if the value was passed in the URL to the Configuration, it will be validated and an error message will appear.
  - This affects both Mobile and Desktop.

"Update and Back" button in Configuration Rule Editor performs as expected (SFDC Bug ID: 22899)
In Version 11.1, when a user clicked "Update and Back" in a Configuration Rule Editor, they would be returned to Model Configuration instead of the previous page. Now, clicking the "Update and Back" button will return the user to a Rule List page within Configuration.

Config attributes and Quick Links appear in the same order (SFDC Bug ID: 27085)
In Version 11.1, the order of Configuration attributes differed from the order that they appeared in Quick Links. Now, Configuration attributes and Quick Links attributes appear in the same order.

Inactive tab attribute values will appear in Pipeline Viewer (SFDC Bug ID: 27725)
In Version 12, if a user set a tab to Inactive in order to hide its attributes, the attribute would appear with blank values in the Pipeline Viewer. Now, Inactive tab attributes will have their proper values in the Pipeline Viewer.

Numeric image menu entries with leading zeroes will remain selected after update (SFDC Bug ID: 31868)
In Version 12, numeric image menu entries with leading zeros would not remain selected after a page refresh and could not be used to trigger rules. This occurred in both single-select and multi-select menus. Now, numeric image menu entries with leading zeroes will remain selected after a page refresh.

HTML body element ID displays consistently (SFDC Bug ID: 33144)
In Version 12.1, when a user inspected a HTML body element ID, the body ID would display inconsistently upon page update. This issue has been resolved.

Scroll arrows will remain in the Layout Editor regardless of tab length (SFDC Bug ID: 36992)
In Version 12, when the Layout Editor tabs became too long, the scroll arrows vanished. This issue has been resolved.

"Convert to Advanced Condition" will not inverse the "Not-Contains" operator (SFDC Bug ID: 38629)
In 2013 Winter, the "Convert to Advanced Condition" inverses the "Not-Contains" operator when viewing the BML. This issue has been resolved.

Hiding rules will perform as expected when simple constraints are added (SFDC Bug ID: 38709)
In 2013 Winter, when a user added simple constraints targeting hidden array control attributes, the hiding rule would be stopped and the attributes would be displayed. Now, when a user adds simple constraints targeting hidden array control attributes, the hiding rule will perform as expected and the attributes will be hidden.

Deployed changes to attribute description appear when hovering over help icon (SFDC Bug ID: 39166)
In 2013 Winter, when a user hovered over the user-side help icon after changing an attribute description, the changes would appear without requiring a deployment. This issue has been resolved.

Simple constraints targeting blank ("") array menu attributes will work (SFDC Bug ID: 42385)
In 2013 Winter, simple constraints targeting only blank ("") array menu attributes do not work. Now, simple constraints targeting blank ("") array menu attributes will work.

User will be returned to same point in Quote after correcting configuration issue when line item total exceeds 90 trillion (SFDC Bug ID: 44571)
In 2013 Spring, when the total for one or more line items was greater than 90 trillion, the configuration lost the information related to the Quote and the user was unable to return to the transaction after they had corrected the issue with the configuration. Now, once the configuration issue has been corrected, the user will be able to return to the Quote at the same point.

Action Attributes tied to Recommendation Rules cannot be deleted (SFDC Bug ID: 45028)
In 2013 Spring, Action Attributes could be deleted even if they were tied to a Recommendation Rule. This caused the configuration to be corrupted and migrations could not be performed. Now, users cannot delete Action Attributes that are tied to Recommendation Rules.

CRM INTEGRATION

CRM return forms have been updated (SFDC Bug ID: 17947)
In Version 11, when a user wrote an advanced destination rule, the expected return text references variable forms and visualforce pages would not be used. Now, the variable sf_opportunity_id has been replaced with crmOpportunityId_quote.

Dates imported from SFDC will convert to system date preference (SFDC Bug ID: 34791)
In Version 12, when a user set a non-US date preference and imported a date from Salesforce, the imported date would not be converted to the user's date preference and the following error message appeared, "Month should be between 1 and 12." Now, dates imported from Salesforce will convert correctly according to the System date preference.

PARTS

Simple Search will be accessible from within Advanced Search (SFDC Bug ID: 18267)
In Version 11, when a user entered Advanced Search from the Parts Search Panel page, there was no option to return to Simple Search. Now, users can return to the Simple Search from within the Advanced Search.

Parts Association and Parts Display pages will utilize the correct Document Mode (SFDC Bug ID: 28248)
Post-2013 Autumn Release, the document mode on Part Association and Parts Display pages would be set to "IE9 Standard" to conform with the Configuration page's document mode. This was previously set to "Quirks" mode.

Parts Integration Scheduler will use user preference time zone (SFDC Bug ID: 42517)
In Version 12.1, when a superuser with a partner login accessed the Parts Integration Scheduler and selected an integration schedule, the scheduled integration would not occur at the expected time. This would occur because the user profile time zone and Schedule time zone were set to different time zones. Now, the scheduling page will use the same time zone as the user preference.

Parts integration may be scheduled for 31-40 minutes (SFDC Bug ID: 42518)
In 2013 Winter, when a superuser with a partner login scheduled a parts integration, selecting 31-40 minutes was not possible. Now, when a superuser schedules a parts integration, 31-40 minutes may be selected.

"last_integration_time" type will be specified (SFDC Bug ID: 42919)
In 2013 Spring, the type would not be specified for "last_integration_time". Because of this, the web service descriptor consumer could not be read. Now, the type will be specified for "last_integration_time".

Parts Import description on Parts Integration page is accurate (SFDC Bug ID: 43889)
In 2013 Spring, the description detailing how Parts Import works on the Parts Integration page was inaccurate. The last sentence read, "Note that this will delete all the existing parts in BigMachines and replace them with the parts from the partner." The Delta Part Import property actually determines whether the parts will be deleted. Now, the last sentence reads, "Note that this will either delete or update all the existing parts in BigMachines depending on the delta part import property."

Can view XML of part added as subnode into "Enter Exact Part Number" field (SFDC Bug ID: 46791)
In 2013 Spring, when a user clicked "Display XML" to view a part added as a subnode into the "Enter Exact Part Number" field, a 500 Error would appear. Now, a user can view the XML of a part added as a subnode into the "Enter Exact Part Number" field.

PRICING

Display of non-USD currency symbols can be turned off (SFDC Bug ID: 19177)
In Version 11, when a user attempted to disable the appearance of their non-USD currency symbol (display_base_currency_symbol) on their profile, the currency symbol continued to remain active. Now, when a user opts not to display their non-USD currency symbol, all currency symbols will be turned off.

DOC ENGINE

Date formatting functions run on empty dates will return nothing (SFDC Bug ID: 21091)
In Version 11, when a user called a date formatting function (e.g. shortdate or longdate) on an empty date, an incorrect string (NaN/NaN/) would be displayed. Now, if the date is empty, nothing will be displayed.

Attached PDF total page count will account for all pages (SFDC Bug ID: 29935)
In Version 12, when a user attached a PDF to the end of a Document using the standard method, only the first page was counted in the PDF's total page count. Now, every page of the attached PDF will be accounted for in the total page count. This functionality is only supported on sites with Antenna House; Apache functionality remains the same.

REPORTING

Data Cube Export mapping files will match .dat file (SFDC Bug ID: 26294)
In Version 12.2.4, when a user runs a Data Cube Export using a mapping file that contains _config_attributes and _config_attr_text, the mapping file contains the deprecated attributes, but the .dat file does not. Now, the deprecated attributes will not appear in the mapping file.

Corrupt transactions will be logged and repopulation process will succeed (SFDC Bug ID: 41022)
In 2013 Winter, when a user repopulated a Data Column within the Commerce Deployment Center, a corrupt transaction would cause the entire process to fail. Now, a corrupt transaction will not stop the entire repopulation process; the corrupt transaction IDs will be logged and the process will succeed.

Reports will be returned when a filter is added to a line level Date Type Attribute (SFDC Bug ID: 42382)
In 2013 Spring, when a user added a filter to a line level Date Type Attribute, an error message appeared and the Report was not returned. Now, when a user adds a filter to a line level Date Type Attribute, a report will be returned.

WEB SERVICES

"Get" action performs as expected using a 1 million-row Data Table (SFDC Bug ID: 33143)
In Version 12.1, when a user performed a "Get" action within Web Services using a 1 million-row Data Table, the site would run out of memory. This issue has been resolved.

Add/update table WSDL contains all the expected fields (SFDC Bug ID: 45026)
In 2013 Spring, the BigMachines Web Services for adding or updating tables contained two fields in the SOAP response that were not found in the WSDL. These two fields were: "records_inserted" and "records_updated". Now, the WSDL contains all the same fields found in the BigMachines add/update tables Web Services SOAP response.

Attribute name search in WSDL will return List, Union, and Restriction (SFDC Bug ID: 48437)
In 2013 Spring, when a user searched the WSDL for an attribute name, the result did not contain List, Union, or Restriction. Now, when a user searches the WSDL for an attribute name, the result will contain List, Union, and Restriction.

BML

getPartData Price Books can be consistently enabled using BML (SFDC Bug ID: 34750)
In previous versions, the getPartData function behaved inconsistently when being specified using BML. Now, the getPartData Price books specifications will behave consistently when using BML.

Price attribute line items appear as expected when printing statements from Commerce BML Editor (SFDC Bug ID: 43878)
In 2013 Spring, when a user attempted to print statements from a Commerce BML Editor with a line_process attribute, the values for the price attributes line items were being printed as 0.0. Now, when a user prints statements from a Commerce BML Editor with a line_process attribute, the price attribute line items will appear as expected.

Sort function will order alphabetical characters correctly (SFDC Bug ID: 47670)
In 2013 Spring, the Sort function would not order alphabetical characters correctly; for example, "Z, a, B, A" would sort as "A, B, Z, a". Now, the Sort function will order alphabetical characters correctly (e.g. "Z, a, B, A" will sort as "A, a, B, Z").

CURRENCY

Currency attributes will be right aligned when added (SFDC Bug ID: 41020)
In 2013 Winter Update 2, when a user added a currency attribute, the currency attribute would align to the left rather than the expected right alignment. Now, when a user adds a currency attribute, the attribute will align to the right.

BULK UPLOAD/DOWNLOAD

Records returned in CSV export will match number of records returned in Search (SFDC Bug ID: 41289)
In 2013 Winter, when a fullaccess_user exported quotes from the Quote Manager, the number of records in the CSV output is nearly double the amount of records that are returned from the Search. Now, the number of records in the CSV file will match the number of records returned from the Search.

DATA TABLES

Known Behavior:

- It is strongly recommended that when using the Data Table Natural Keys feature, introduced in 2013 Spring, that no more than five string columns be assigned to a part of the key.
- If more than six string columns are assigned to the Natural Key, an error message appears ("The call failed on the server; see server log for details") and the columns marked as keys will not display correctly.
  - To fix this issue, admin must remove the Natural Keys from all columns and Save the Data Table.
    - Once this is done, Natural Keys can be added to six or fewer columns.

Data Table Manager loads tables in a timely manner (SFDC Bug ID: 42225)
In 2013 Winter Update 2, the Data Table Manager would take an excessively long time to load. This issue has been resolved.

Data Table float column precision is set at six decimal places (SFDC Bug ID: 45674)
In 2013 Spring, Data Table float columns could hold up to nine decimal places, but when a value with more than six decimal places was entered, the value cleared and reverted to the previous state. Now, Data Table float columns are set to hold up to six decimal places; if more decimal places are entered, the value will be truncated.

Searches will utilize the "like" function instead of "equals" (SFDC Bug ID: 46878)
In 2013 Spring, when a user performed a search, the search results returned values that were "equal" to the search, not "like". This caused fewer search results to be returned. Now, when a user performs a search, the "like" function will be used to return results.

Searches for String values will not return Integers and Floats (SFDC Bug ID: 49032)
In 2013 Spring, when a user searched across tables for a String value, rows that contained columns with Integers and Floats would be returned (in addition to actual String values). Now, Integers and Floats will not be returned when a search for String values is performed.

FORMULA MANAGEMENT

Default output will be saved in Formulas (SFDC Bug ID: 43074)
In 2013 Spring, the default output would not be saved in Formulas. Now, the default output will be saved in Formulas.

Both display and variable values will be filtered when searching for an attribute (SFDC Bug ID: 43535)
In 2013 Spring, when a user searched for an attribute, only the display values would be searched but the values in both the display and variable would be highlighted. Both display and variable values should be searched. Now, when a user searches for an attribute, both the display and variable values will be searched.

Advanced modification float attributes will round to the correct number of decimal places (SFDC Bug ID: 43964)
In 2013 Spring, quote float attributes within an advanced modification could not exceed two decimal places. Floats exceeding two decimal places would be rounded to two decimal places. Now, all float attributes will be rounded to the correct number of decimal places before being used in Formulas.

INTERNATIONALIZATION

Convert to Advanced Text will be available in Text Administration (SFDC Bug ID: 18266)
In Version 11, the "Convert to Advanced Search…" text on the Parts Search Panel page would not be available for modification in Text Administration. Now, the "Convert to Advanced Text" will be available in Text Administration.

Host Company page will retain newly added language after time zone update (SFDC Bug ID: 43534)
In 2013 Winter Update 4 and 2013 Spring, when a superuser changed the time zone for the Host Company after adding a new language, the newly added language would not appear on the Host Company page. Now, newly added languages will remain on the Host Company page after the time zone is updated.

FILE MANAGER

File Manager modsecurity will behave as expected (SFDC Bug ID: 32033)
In Version 12.1, if a user uploaded a file that had a name beginning with "ps" or "cc" to a folder (that was not the default folder), a modsecurity error would prevent any new files from being added to that folder. Now, uploaded files will behave as expected; modsecurity has been updated to allow modifications to the File Manager.

Users must be logged into site to access files within the File Manager (SFDC Bug ID: 41286)
In 2013 Winter Update 2, when a user turned on the Secure File Manager, files could still be accessed without logging onto the site by using a bmfsweb path. Now, the Secure File Manager ensures that the user accessing files stored in the BMI File Manager is logged into the site.

GENERAL

The calendar tool has been replaced. The new tool functions in the same manner as the previous calendar tool.

- The new calendar tool reflects user time preference.
  - Time preference is set here: My Profile > Preferences > Date/Time Format.

Warning messages will appear when migration of group restricted data columns fail (SFDC Bug ID: 43882)
In 2013 Winter Update 3, when a user migrated a data column with a restriction based on a group, a warning message would appear in the logs, regardless of success or failure. Now, when a user migrates a data column with group-based restrictions, warning messages will only appear if the migration fails.

SiteMetrics Usage Reports will return the specified date values (SFDC Bug ID: 43883)
In 2013 Spring, when a user executed a SiteMetrics Usage Report using a specific date selection, the report was returned using incorrect date values. Now, when a user executes a SiteMetrics Usage Report using a specific time period, the report will return using the correct date values.

Monthly Usage Report "User (Most recent login)" columns will appear as expected (SFDC Bug ID: 46560)
In 2013 Spring, when an active user logs into BigMachines for the first time and views the "User (Most recent login)" column in the Monthly Usage Report, the following appears, "Firstname Lastname (>6 months)." Now, when an active user logs into BigMachines for the first time and view the "User (Most recent login)" column in the Monthly Usage report, the following will appear, "Firstname Lastname (N/A)."

Text Administration Key will perform as expected (SFDC Bug ID: 49163)
In 2013 Spring, when a user clicked on the Text Administration Key, "Your password has expired. Please change your password or contact the System Administrator to reset," a "Status 500" internal server error would occur. Now, the Text Administration Key, "Your password…." will perform as expected.

UPGRADING TO BIGMACHINES 2013 AUTUMN

All test instances should be upgraded and tested on BigMachines 2013 Autumn before upgrading the target sites.

BROWSER SUPPORT

When using standard functionality, 2013 Autumn is compatible with:

Desktop:

User-Side: IE8, IE9, IE10, FF20.x, and Chrome 26.x
Admin-Side: IE8 and FF20.x

Mobile:

iOS 6: Safari 6.x
Android 4.1 Jelly Bean: Chrome 26.x

Note: Sites created with additional JavaScript, alternate CSS, or other custom functionality, may no longer be compatible with our supported browsers. Please contact Customer Support to determine if workarounds and minor fixes are available. Add-on work may be required to support customizations.

TRAINING

Please refer to release notes for all versions between your current version and the version that you are upgrading to in order to see all new functionality and known issues.
It is strongly recommended that users refer to the help documentation to become familiar with the new features introduced in BigMachines 2013 Autumn. For additional help, please see the BigMachines Support Center.
Any information not explicitly mentioned in the release notes as being supported by the software can be verified against the product help for BigMachines 2013 Autumn and/or with the BigMachines Professional Services team.

DISCLAIMER

These Release Notes are provided for high-level informational purposes only and are not intended to function as a specification or to replace the Product Help and/or User Guide. Upgrading to a subsequent release may require commerce processes, configuration, and/or global function settings to be re-deployed. Modifications to integration XML and/or APIs also may be required.

UNPLUGGED

As of 2013 Autumn, BigMachines will no longer support Unplugged functionality.