

Oracle® Cloud

Designing with Financial Reporting Web Studio for Oracle Planning and Budgeting Cloud

Oracle Cloud Designing with Financial Reporting Web Studio for Oracle Planning and Budgeting Cloud

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Documentation Accessibility	9
Documentation Feedback	11
Chapter 1. Designing Reports	13
Before Designing a Report	13
Design Considerations	14
Report Performance	14
Data Queries - Expanded Versus Single Data Segments	14
Suppressing Missing Blocks	14
Renaming Dimensions and Members	15
Designing Dynamic Reports	15
Using Financial Reporting Web Studio to Design a Report	15
Working with Report Objects	16
Adding Report Objects	16
Positioning Report Objects	17
Resizing Report Objects	17
Defining Properties for Report Objects	18
Inserting Page Breaks Before Report Objects	18
Inserting Saved Report Objects	18
Defining the User Point of View	19
About the User Point of View	19
Setting Up the User Point of View	19
Creating Headers and Footers	20
Creating a Title Page	20
Using Text Functions to Display Information	20
Searching for Information in the Repository	21
Viewing Report Summaries	21
Viewing Report Server Information	21
Chapter 2. Working With Grids	23
About Grids	23
Adding a New Grid	23

Inserting an Existing Grid	24
Selecting a Grid	24
Defining the Grid Point of View	25
About the Grid Point of View	25
Setting Up a Grid Point of View	25
Showing or Hiding the Grid Point of View	26
Selecting Members for the Grid Point of View	26
Working With Rows and Columns in Grids	27
Inserting Rows and Columns	27
Changing the Row Height and Column Width	28
Defining Row and Column Headings	28
Overriding a Row or Column Heading	29
Cutting, Copying, and Pasting Rows and Columns	29
Inserting Page Breaks Before a Row or Column	30
Defining Formula Rows and Columns	30
Expanding Rows and Columns	33
Formatting Grids	37
Expanding the View of a Grid	37
Formatting Numbers in Grids	38
Using Special Currency Formats	38
Aligning the Text in a Grid	39
Hiding Rows, Columns, or Grids	40
Modifying the Font	40
Creating Custom Colors	40
Adding Borders and Shading to Cells in Grids	41
Freezing Grid Headers	41
Inheriting Cell Formatting	41
Copying Cell Formatting	42
Merging Several Cells Into One Cell	43
Using the Formula Bar	43
Creating Formulas	43
Deleting a Formula Row or Column	44
Modifying Member Selection Using the Formula Bar	44
Example Report with Formulas	45
Setting Data Query Optimization Options	46
Showing Supporting Detail for Planning	47
Chapter 3. Working With Text Objects	49
Adding Text Objects	49

Adding a New Text Object	49
Inserting an Existing Text Object	50
Formatting Text Objects	50
Defining Fonts for Text	50
Adding Borders and Shading to Text	50
Aligning Text	50
Chapter 4. Working With Images	53
Inserting Images	53
Formatting Images	54
Chapter 5. Working With Charts	55
About Charts	55
Inserting a Chart	56
Defining Chart Data	56
Defining Chart Titles, Legends, and Labels	57
Defining Chart Titles	57
Defining Chart Legends	57
Defining Chart Axis Titles	58
Defining Axis Labels	58
Formatting Charts	60
Formatting Fonts	60
Formatting Borders	60
Formatting Bars in Bar and Combo Charts	61
Formatting Lines in Line and Combo Charts	61
Formatting Combo Charts	62
Formatting Pie Charts	62
Formatting Background and Gridline Options	62
Defining Styles for Data Sets	63
Positioning Charts in Reports	63
Resizing and Moving Charts	63
Chapter 6. Defining Members	65
About Defining Members	65
Assigning Members	65
Assigning Members to Data Rows or Columns	66
Assigning Members to the Page Axis	66
Defining Special Types of Members	67
Assigning Members Using Functions	68
Searching For Members	70

Previewing Selected Members	71
Copying Members Names From a Text Application	71
Removing Members	71
Selecting Multiple Members	72
Access Privileges for Members in the Planning Database Source	73
Chapter 7. Formatting Reports	75
Formatting Best Practices	75
Changing the Page Setup	75
Chapter 8. Providing For Detailed Data and Documents in Reports	77
Setting Up Expansions to Access Detailed Data in Reports	77
Positioning Expansion Rows and Columns	77
Attaching Cell Documents in a Report	78
Supported Microsoft Office File Types	78
Viewing Reports Containing Cell Documents	79
Displaying a List of Cell Document	79
Chapter 9. Previewing and Printing Reports	81
Previewing Reports	81
Responding to Prompts When Previewing	81
Printing Tips for Designers	82
Positioning and Alignment	82
Printing Text Boxes	82
Printing Grids	83
Printing Page Dimension Headings	83
Printing Headers and Footers	84
Chapter 10. Using Functions	85
Mathematical Functions	86
About Mathematical Functions	86
Abs	92
Average	93
AverageA	94
CountA	95
Difference	95
Eval	96
Max	97
Min	98
PercentofTotal	98
Product	99

Round	100
Sum	101
Truncate/Truc	101
Text Functions	103
About Text Functions	103
Annotation	108
CellText	110
Date	113
DataSource	115
Footnote	116
GetCell	118
GetHeading	119
GridDimension	120
ListofCellDocuments	121
MemberAlias	122
MemberName	123
MemberProperty	124
Page	126
PageCount	126
PlanningAnnotations	126
ReportAuthor	128
ReportCreated	129
ReportDesc	129
ReportFolder	130
ReportModified	130
ReportModifiedBy	131
ReportName	131
ReportRunBy	131
RetrieveValue	132
Conditional Functions	132
IFThen, If	132
Conditional Operators	132
Complex Conditions	134
Financial Functions	135
Rank	135
Variance/Var	137
VariancePercent/VarPercent	139

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Documentation Feedback

Send feedback on this documentation to: epmdoc_ww@oracle.com

Follow EPM Information Development on these social media sites:

LinkedIn - http://www.linkedin.com/groups?gid=3127051&goback=.gmp_3127051

Twitter - <http://twitter.com/hyperionepminfo>

Facebook - <http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642>

Google+ - <https://plus.google.com/106915048672979407731/#106915048672979407731/posts>

YouTube - <https://www.youtube.com/user/EvolvingBI>

1

Designing Reports

In This Chapter

Before Designing a Report	13
Design Considerations.....	14
Using Financial Reporting Web Studio to Design a Report.....	15
Working with Report Objects	16
Defining the User Point of View	19
Creating Headers and Footers	20
Creating a Title Page.....	20
Using Text Functions to Display Information.....	20
Searching for Information in the Repository	21
Viewing Report Summaries.....	21
Viewing Report Server Information	21

Before Designing a Report

Before designing a report, consider:

- What is the goal of the report?
- Who are the end users of the report?
- How frequently is the report generated and by how many users?
- What is an acceptable retrieval/generation time?
- How many cells does the report query before suppression?

Design Considerations

Subtopics

- [Report Performance](#)
- [Data Queries - Expanded Versus Single Data Segments](#)
- [Suppressing Missing Blocks](#)
- [Renaming Dimensions and Members](#)
- [Designing Dynamic Reports](#)

Report Performance

Report performance can vary for each user. What is acceptable for one user may not be acceptable for another. For example, if reports are run in batches during off-peak hours, they may have to be completed before the next business day. When data is updated hourly, report performance may only be acceptable if report output returns in seconds.

Data Queries - Expanded Versus Single Data Segments

A data segment is a row or column that retrieves data from a database. An expanded data segment is a row or column that can expand, so that the resulting grid expands to two or more rows or columns when viewed. Often, expanded data segments use functions such as *Children Of* or *Descendants Of*. A single data segment is a row or column that remains a single row or column when shown in the viewer.

While it is generally valid to use expanded and single data segments in the same grid, when designing a grid with large amounts of data, consider using expanded data segments rather than single data segments. Expanded data segments provide some performance advantages over single data segments; however, to produce detailed formatting on different data rows or columns, use single data segments.

Suppressing Missing Blocks

Note: The ability to suppress missing blocks is available for Oracle Hyperion Planning users only.

You can suppress missing blocks to improve performance when rows or columns contain sparse dimensions. When you suppress missing blocks, you can place large sparse dimensions in rows, while providing good response time if the density of the query is low. Only blocks with data are retrieved. For example, if you place an employee dimension consisting of thousands of members in rows, and place the entity in the page or POV, then only employees of the selected entity are retrieved.

Suppressing missing blocks aids in suppressing missing data when a large number of rows, such as 90% or more, is suppressed. Suppressing missing blocks can degrade performance, however,

if few or no rows are suppressed. Also, certain suppressed blocks may ignore dynamic calculation members.

Renaming Dimensions and Members

If you rename the dimensions or members in the data source, you must manually update each report in Oracle Hyperion Financial Reporting Web Studio to reflect the changes.

Designing Dynamic Reports

When designing dynamic reports:

- Use expanded data segments for optimal performance:
 - Use functions on expanded data segments that are not placed on separate rows or columns.
 - Use multiple member selections on expanded data segments that are not placed in separate rows or columns.
 - Use single data segments only when required for formatting or calculations.
- Write efficient formulas:
 - Use row or column formulas rather than cell formulas, when possible.
 - Use reference properties.
 - Use a cell reference instead of cross-axis references.
 - Remove unnecessary parenthesis from formulas.
- Do not limit your report to the Grid object:
 - Add functions in text boxes that highlight specific areas.
 - Hide a grid of data in a report to highlight just the graphics.
- Create report templates:
 - Leverage the power of POVs and prompting functions.
 - Design once using the Current POV and prompting functions.
 - Learn the power of Grid, User, Book, and Batch POVs.
- Create reusable objects and row and column templates.

You can use Text, Image, Grid, Chart objects, and row and column templates in multiple reports.

Using Financial Reporting Web Studio to Design a Report

You design reports using a blank report layout where you can specify the contents of the report, including grids, charts, images, text boxes, headers, and footers.

► To design a report:

1 In **Financial Reporting Web Studio**, click 

2 Design the report in the **Report** tab.

Working with Report Objects

Subtopics

- [Adding Report Objects](#)
- [Positioning Report Objects](#)
- [Resizing Report Objects](#)
- [Defining Properties for Report Objects](#)
- [Inserting Page Breaks Before Report Objects](#)
- [Inserting Saved Report Objects](#)

Report objects include:






- Grids
- Text
- Images
- Charts

When you add report objects to a report, you define their position in the report layout, and you select the data content for the report object. You can also insert report objects previously stored in the repository. You can use report objects in multiple reports.

After you add a report object to a report, you can assign properties to it such as format options. Format options include font, alignment, spacing, shading, and positioning.

Adding Report Objects

To add a report object to a report, do one of the following:

- Click the icon for the report object    , and then draw an outline for the object in the desired position in the **Report** tab.
- Select **Insert**, then **Grid**, **Text**, **Image**, or **Chart**, and then draw an outline for the object in the desired position in the **Report** tab.
- In the **Report** tab, in the **Header**, **Body**, or **Footer** section, click , and then select the report object to add. (This draws a fixed size object in a pre-determined location in the Report Designer. You can then re-size or move the object if desired.)

Tip: If you add a report object in one of the sections (Header, Body, or Footer) in the Report Designer, and you want to add another report object in a different section, you must first click the name of the report (above the Header, Body, and Footer) to get back to the “layout view.” You can then go to another section in the Report Designer and add another object.

Positioning Report Objects

Positioning report objects involves placing objects in the desired positions as you design the report in Financial Reporting Web Studio. For example, you can position a text box on the top left of the report, and position a grid in the center of the report. How you position objects affects the online report and the printed page.

It is important to position report objects because the objects can change size depending on the amount of data they contain.

You can set the position of a report object relative to other report objects. For example, if report objects above a specific report object increase or decrease in size, the specific report object moves as necessary to maintain the same distance between it and the other report objects. If you do not position a report object, the report object is printed at its current location regardless of whether other report objects increase or decrease in size.

Overlapping objects enables you to organize the objects when laying out a report. When you overlap objects, you can send a report object to the back (behind another object) or bring it to the front (ahead of another object).

When working with multiple overlapped objects, you can print the report regardless of your object layout. For example, if a portion of a text box overlaps a chart, or if layers of charts are placed on top of one another, you can still print the report. Keep in mind, however, that the overlapping objects may not print as displayed in Financial Reporting Web Studio, and you may need to rearrange the objects as you design your report.

To position a report object, do one of the following:

- Click the report object and drag it to the desired position.
- Select the report object, then in the object’s **Properties**, under **Position**, select the object’s horizontal and vertical position in the report.

Resizing Report Objects

You can resize report objects so that they are displayed fully expanded or at a specified height and width when you preview or print a report. When you resize report objects, the content remains the same size, but the space around the contents changes. For example, you can drag a grid border to reserve an amount of space to the left of the grid. When you view the grid, the area to the left of the grid remains blank.

In addition to resizing report objects, you can design reports to automatically increase the size of text objects and grids as more area is needed to include all resulting data.

► To resize a report object:

- 1 Click the name of the report that contains the object to return to the report “layout view”.
- 2 Click the object.

The object’s border is highlighted in red to indicate that the object is selected.

- 3 Click on the border of the object and drag it to change the size.

► To automatically resize text objects and grids:

- 1 Select the text object or grid in the section where it is inserted.

You can insert text objects in the Header, Body, or Footer sections. You can insert grids only in the Body of a report.

- 2 In **Text Properties** or **Grid Properties**, select **Autosize**.

Defining Properties for Report Objects

When you select a report object, the corresponding properties for the object are displayed on the right side of the design area. For example, when you select a grid, the **Grid Properties** are displayed. Properties allow you to set options for the report object.

Inserting Page Breaks Before Report Objects

You insert a page break before a report object to move the report object to the next page.


► To insert a page break before a report object:

- 1 In the **Report** tab, select a report object.
- 2 In the report object’s properties, select **Page Break Before**.

Inserting Saved Report Objects

You can insert saved grids, text, images, and charts from the repository into a report.

► To insert a saved report object:

- 1 From the **Report Designer**, do one of the following.
 - Click in a section in the Report Designer, then select **Insert**, and then **Saved Object**.
 - Right-click in a section in the Report Designer, and select **Insert Object**, and then **Saved Object**.
 - In the **Body** section in the Report Designer, click , and then select **Saved Object**.
- 2 In the **Insert Saved Object** dialog box, specify the object to insert.
- 3 Click **Insert**.

Defining the User Point of View

Subtopics

- [About the User Point of View](#)
- [Setting Up the User Point of View](#)

About the User Point of View

Every data value in a report is derived from the intersection of a member from each dimension in an Planning database connection. In Financial Reporting Web Studio, you can place these dimensions on the report grid or in the user point of view. When you view the report, you can change the member selected for dimensions on the user point of view. This allows you to customize a report to fit your needs. You can also use the user point of view in books.

In a report, the member specified for a dimension in the user point of view is used for all grids with that dimension. For example, if you select the “budget” member for the “scenario” dimension, all grids in the report with “scenario” in the user point of view use “budget” when you retrieve data. The user point of view is not displayed if all of the dimensions are placed on the grid.

The following dimensions are displayed on the user point of view bar:

- Dimensions not currently defined in a row, column, or page
- Dimensions in a row, column, or page that are flagged for the current point of view
- Dimensions without a member selected on a grid point of view
- Attribute dimensions dragged to the point of view area in the Dimension Layout dialog box

Setting Up the User Point of View

When you design a report, you limit the members on the user point of view for a database connection. The user point of view applies to all grids that use that database connection.


➤ To set up the user point of view:


1 With a grid selected, select **Task**, and then **Set Up User POV**.

2 In the **Setup User POV** dialog box:

- For each dimension, define whether the user point of view will include all members in the dimension, or an individual member.

The default is to include all members for each dimension. To define an individual

member for a dimension, click , and then select a member from the **Select Members** dialog box. If you have an individual member selected and wish to return to


the setting to include all members, click .

- Select **Show Dimension Name** to show the dimension name in the user point of view.
 - Select **Show Member Name** to show the member name in the user point of view.
 - Select **Show Alias** to show alias names in the user point of view. If you select Show Alias, select an option for how to display the alias.
- 3 Click **OK** to save the information for the user point of view.

Creating Headers and Footers

Headers and footers can be used to add a description, add graphics, or insert text functions. The header is located immediately below the top margin, and the footer is located immediately above the bottom margin.

► To create a header or footer:

- 1 With a report displayed in Financial Reporting Web Studio, in **Report**, in the **Header** or **Footer** section, click , and then select whether to enter a Saved Object, an Image, or Text.
- 2 Enter the **Header Height** or **Footer Height** in inches.
- 3 Customize the header or footer by entering additional properties.

Creating a Title Page

► To create a title page:




- 1 Open a report to which you want to add a title page and add a text box in the **Header** section in the Report Designer workspace.
- 2 Select a report object below the text box, then from that report object's property sheet, select **Page Break Before**.
- 3 Design the text for the title page using the options in the Text Properties sheet.

Using Text Functions to Display Information

Text functions can display report information such as the report name or description, information about a database associated with a grid, or information about data in your database connection.

You can use text functions in text boxes, data rows or columns, formula rows or columns, or page headings. Enclose the formula in double angle brackets (<<) and (>>) to distinguish different text functions. The font for the text function can be formatted like regular text. See [“Text Functions” on page 103](#) for information about text functions.

- To use text functions to display information in a report:
- 1 With a report open in the Report Designer, do one of the following

- Select a text box, and then click .
- Select a text cell in a grid, and then in **Cell Properties**, click .
- Select a heading cell in a grid, and then in **Heading Column Properties** or **Heading Row Properties**, select **Custom Heading** and click .


- 2 In the **Insert Function** dialog box, select a function, and then click **OK**.


Note: You can enable or disable error messages for text functions that are displayed in the grid or text object. The default is to display error messages. To disable this feature, from **Text Properties**, clear **Show Text Function Errors**.

Note: You must format the entire text function, including the angle brackets, with the same font properties.

Searching for Information in the Repository

- To search for information in the Financial Reporting repository:

- 1 Open **Financial Reporting Web Studio**.
- 2 In the **Repository** section, enter a text string to search for, and then click .

Tip: To clear the results of the search, click .

Viewing Report Summaries

A report summary shows the name, the description, and the creator of a report. In addition, a report summary shows the modifier, date, and time the report was last modified. Summary information is saved when you save the report.

To view report summaries, select **File**, and then **Summary**.

Viewing Report Server Information

You view report server information to monitor the activity on your server.

- To view server information:

- 1 Open a report.

2 Select **File**, and then **Report Server Info**.

2

Working With Grids

In This Chapter

About Grids	23
Adding a New Grid	23
Inserting an Existing Grid	24
Selecting a Grid	24
Defining the Grid Point of View	25
Working With Rows and Columns in Grids	27
Formatting Grids	37
Using the Formula Bar	43
Setting Data Query Optimization Options	46
Showing Supporting Detail for Planning	47

About Grids


Grids are tables that can contain data from external database connections. You can add a grid to your report, define its dimension layout, select members, and then format the grid. You can also insert an existing grid to reuse a pre-formatted grid design. You can use text, dimensions, members, and formulas to define the grid content. You can also hide a grid. For example, you might want to display a chart in a report, but not the grid associated with the chart.


Note: In Microsoft Excel, reports containing two grids that are side by side do not display the grids side by side; instead, the grids are displayed one above the other.

Adding a New Grid

► To add a new grid to a report:

1 Do one of the following:

- Click , and then position the grid in the **Body** section in the Report Designer.
- Select **Insert**, then **Grid**, and then position the grid in the **Body** section in the Report Designer.

- In the **Body** section in the Report Designer, click , and then select **Grid**.
- 2 In **Database Connection Properties**, select an existing database connection, or click **New Connection** to create a new database connection.
 - 3 In **Dimension Layout**, drag dimensions from **Attribute Dimension** (if they exist) and from **Point of View**, into **Page, Rows, and Columns**, and then click **OK**.


You can place multiple dimensions into Page, Rows, and Columns. The dimensions that are not placed in Pages, Rows, or Columns are displayed in the user point of view bar located at the top of the Report Designer for the grid.

You can layer multiple dimensions in a row, column, or page. For example, you might want to place the Scenario and Time dimensions on the same axis to show actual and budget data over a period of time.

When you add a new grid, the system assigns a default name to the grid. The default name assigned is Grid n , where n is a system-assigned identification number. For example, if you save a grid that is the sixth system-named grid saved in the report, the default name is Grid6. You can rename the grid in **Grid Properties**.

The system uses the grid name when a function or chart references the grid. For example, if you design a chart to graphically display the data from a grid, the chart properties must reference the grid by its name.

Inserting an Existing Grid

- To insert an existing grid in a report:
- 1 In the **Body** section in the Report Designer, do one of the following:
 - Select **Insert**, and then **Saved Object**.
 - Right-click and select **Insert Object**, and then **Saved Object**.
 - Click , and then select **Saved Object**.
 - 2 In **Type**, select **Grid**.
 - 3 Select the desired grid object, and then click **Insert**.

Selecting a Grid

To select a grid in the Report Designer, click the top left cell to select the rows and columns inside the grid.

When you select a grid, you can right-click to access the following options:

- **Dimension Layout**—Change the dimensions for the pages, rows, and columns.
- **Grid Point of View**—Display or hide the Grid Point of View bar.

- **Grid Point of View Setup**—Select members for the Grid Point of View.
- **Data Query Optimization Settings**—Suppress the rows on the database connection server or the report server. (See “[Setting Data Query Optimization Options](#)” on page 46.)

Defining the Grid Point of View

Subtopics

- [About the Grid Point of View](#)
- [Setting Up a Grid Point of View](#)
- [Showing or Hiding the Grid Point of View](#)
- [Selecting Members for the Grid Point of View](#)

About the Grid Point of View


The grid point of view (POV) is the user point of view at the grid level. The grid POV enables you to specify the members for a dimension in a grid without placing the dimension on the row, column, or page intersection. This allows you to control or limit the selection of members available to users who view or print the report. Setting the grid POV prevents prompting for the user POV each time the report runs. After you specify a dimension in the grid POV, you can determine how viewers select members.

By default, the grid POV is set to use whatever dimensions are in the user POV. If one or more dimensions are needed to have a selection different than the user POV, you can select the dimension(s) for the grid POV, and the report will use the grid POV instead of the user POV.

Tip: To improve report performance, do not use dimensions with only one member selected for the entire grid in a row, column, or page axis. Although it does not increase the potential cell count, it adds overhead to the entire query. Dimensions that are substitution variables or set to one member should be left in the POV. For point of view dimensions where the member can be changed by the end user, use the user POV. For point of view dimensions where the member will remain fixed at a specific selection, use the grid POV.


Setting Up a Grid Point of View


➤ To set up a grid point of view:

- 1 In the Report Designer, select the top left cell of the grid to select the entire grid.
- 2 Do one of the following:
 - Right-click the grid, and then select **Grid Point of View Setup**.
 - In **Grid Properties**, click .
- 3 In the **Setup Grid POV** dialog box:

- For each dimension, define whether the grid point of view will include all members in the dimension or an individual member.

The default is to include all members for each dimension. To define an individual

member for a dimension, click , and then select a member from the **Select Members** dialog box. If you have an individual member selected and wish to return to

the setting to include all members, click .

- Select **Show Dimension Name** to show the dimension name in the grid point of view.
- Select **Show Member Name** to show the member name in the grid point of view.
- Select **Show Alias** to show alias names in the grid point of view. If you select Show Alias, select an option for how to display the alias.

4 Click **OK** to save the information for the grid point of view.

Showing or Hiding the Grid Point of View

As you design your reports, you can hide or display the grid point of view. Hiding the grid point of view prohibits you from changing the grid point of view when viewing a report. By default, the grid point of view is hidden. However, even when the grid point of view is hidden, its values are still operative when running the report.

To show or hide the grid point of view, do one of the following:

- Select the grid, and then in **Grid Properties**, select or clear **Grid Point of View**.
- Right-click the grid, and then select **Grid Point of View**.

Note: The grid point of view displays only in HTML view. Since the report is interactive in HTML view, you can change the grid point of view members. Since PDF documents are static documents, you cannot change the grid point of view in PDF documents.

Selecting Members for the Grid Point of View

On the grid POV bar, you can select members for dimensions that are not used in rows, columns, or the page axis. The members defined on the POV bar complete the criteria used to retrieve data for the grid. Each POV dimension can contain only one member.

Note: Using dimensions with security on the grid point of view may cause unexpected results.

► To select a member for the grid point of view

1 Display the grid point of view bar.

- Select the grid, and then in **Grid Properties**, select **Grid Point of View**.
- Right-click the grid, and then select **Grid Point of View**.

- 2 On the grid point of view, click a dimension to display the **Select Members** dialog box.
- 3 In the **Select Members** dialog box, select a member for each dimension, and then click **OK**.

Working With Rows and Columns in Grids

Subtopics

- [Inserting Rows and Columns](#)
- [Changing the Row Height and Column Width](#)
- [Defining Row and Column Headings](#)
- [Overriding a Row or Column Heading](#)
- [Cutting, Copying, and Pasting Rows and Columns](#)
- [Inserting Page Breaks Before a Row or Column](#)
- [Defining Formula Rows and Columns](#)
- [Expanding Rows and Columns](#)

Inserting Rows and Columns

You insert rows or columns into a grid to add data to the grid. Types of rows and columns include:

- **Data**—Members of dimensions that are retrieved from a database connection. The member in a data row or column defaults to the top-level member of the dimension.

A # symbol indicates data rows, columns, or cells.

- **Formula**—Values based on formulas. Commonly used to calculate totals, averages, and variances on data rows or columns.

An = # symbol indicates formula rows, columns, or cells.

- **Text**—Text typed into the cells or added dynamically through a text function.

Text rows, columns, or cells in grids do not contain symbols when added to a grid. They are initially blank until populated with text.

When you add a grid to a report, it contains one of each of the following:

- Data row
- Anchor row
- Data column
- Anchor column

An anchor row or column is a blank row or column located at the end of the rows or columns. You can add one or more rows or columns in a grid, or you can use the anchor row and column to append new rows and columns.

► To insert a row or a column in a grid:

- 1 In an existing grid, select a cell, row, or column.
- 2 Select **Insert**, then **Row or Column**, and then **Data, Formula, or Text**.

To insert multiple rows or columns, select the number of rows or columns on the grid equal to the number of rows or columns that you want to insert.

3 **Populate the row or column:**

- For a data row or column, double-click the row or column heading and select dimension members to place in the row or column. See [“Assigning Members to Data Rows or Columns” on page 66](#).
- For a formula row or column, select the row or column heading and build a formula in the formula bar at the top of the Report Designer.
- For a text row or column, enter text directly in the row or column, in the formula bar at the top of the Report Designer, or in Text Properties. See [“Formatting Text Objects” on page 50](#).

Changing the Row Height and Column Width

You can change row heights and column widths in grids. You can apply these changes to the entire grid or to specific rows or columns.

► To change row height and column width:

- 1 In an existing grid, select row or column.
- 2 In **Row Properties**, enter a **Row Height**.

Select **Adjust Row Height to Fit** to increase the height of the row to fit all the contents of the cells in that row. (Applicable to PDF Preview)

3 In **Column Properties**, enter a **Column Width**.

Select **Adjust Column Width to Fit** to increase the width of the column to fit all the contents of the cells in that column. (Applicable to PDF Preview)

Tip: You can also drag the handle of a row or column in the grid to the desired size. If you make a row or column too small to see, when the pointer becomes a split line, drag the row or column handle until the row or column is visible.

Defining Row and Column Headings

► To define row and column headings for a grid:

- 1 In an existing grid, select the cell in the upper left corner of the grid.
- 2 In **Grid Properties**, under the **Headings** section, define the following information about the heading:
 - In **Member Labels**, select what to display in the heading (**Member Name**, **Alias**, or **Both**).

- In **Row Headings Before**, select a column.

The row heading is positioned to the left of the selected column. Columns are represented alphabetically as Column A, Column B, Column C, and so on. You cannot place row headings before columns that are sorted.

- Select **Suppress Repeats** to hide headings for reporting members in a segment.

When you suppress repeats, only the first instance of the heading for a repeated member is displayed. Headings for additional repeating members are hidden.


- In **Alias Table**, select the alias table to use when showing aliases.

You can select alias tables in your database connection. Although you cannot specify aliases at the grid level, if you display aliases for a row or column, the system displays the alias table as selected at the grid level. The alias table is initially set to the default alias table for the database connection type.

Note: The heading properties you define apply to all rows and columns unless you specify otherwise.

Overriding a Row or Column Heading

➤ To override a specific row or column heading in the grid:

- 1 Select a row or column heading in the grid.
- 2 In **Heading Row Properties** or **Heading Column Properties**, select **Custom Heading**, and then either:
 - Enter the text to use for the heading.
 - Enter a text function, or click  and select the text function you want to use for displaying data on the headings.
- 3 **Optional:** Select **Allow Expansion** to enable expansion of the summary rows and columns under the heading.

Cutting, Copying, and Pasting Rows and Columns

When cutting or copying rows and columns, keep in mind the following points:

- You can copy rows and columns and paste them in the same grid or between two different grids in a report.
- If you cut a row or column, you can only paste it in the same grid.
- For formula rows and columns, you must copy or cut the entire row or column.

To cut, copy, and paste rows or columns in grids, select the desired rows or columns, and then right-click and select **Cut**, **Copy**, or **Paste**.

Inserting Page Breaks Before a Row or Column

Inserting a page break before a row or column in a grid enables you to break the report data at a specified location while keeping required line items together. Page breaks are indicated with a heavy, dashed line.

► To insert a page break before a row or column in a grid:

1 With a grid displayed in the Report Designer, select a row or column.

You cannot insert a page break before the first column in a report or above the first row in a report.

2 In Row Properties or Column Properties, select Page Break Before, and then select where to position the page break:

- For rows, select **Position at Top** to place the remaining part of the grid at the top of the new page. For columns, select **Position at Left** to place the remaining part of the grid in the left margin of the new page.
- Select **Same Position** to place the grid in the same position as it is currently placed. For example, if a grid is placed half way down the page and you insert a page break with **Same Position** selected, the remaining part of the grid will be displayed half way down the second page.

Defining Formula Rows and Columns

Subtopics

- [Formula Row and Column Syntax](#)
- [Inserting a Formula Row or Column](#)
- [Showing the Formula In a Row or Column Heading](#)
- [Calculating at the Cell Level](#)

A formula consists of a combination of grid references, mathematical functions, and arithmetic operators. A grid reference uses values from the current grid or another grid in a calculation. You can define arithmetic formulas on formula rows, columns, cells, or a range of formula cells in a grid. For example, you can build a formula that adds one or more rows. Or, you can multiply two rows, add a third row, then display the result. For an explanation of the mathematical functions you can use in formulas, see [“Mathematical Functions” on page 86](#).

When defining formula rows or columns, keep in mind the following points:

- Formula results are calculated based on the underlying data values in one or more specified cells. If the cells used in a formula are formatted to use scaling, the formula results may differ from the displayed values for the cells, since the underlying, unformatted data values are used in the formula.
- By default, cells that contain missing data (`#MISSING`) are not treated as zero (0) in formulas, unless specified differently using the `“IfNonNumber”` property. In a formula that uses division, an error is returned.

- Use dot notation to specify mathematical properties. Dot notation is a syntax that specifies properties for a grid or another property. Specify the property using a period (.) followed by the property name.

Formula Row and Column Syntax

Syntax:

gridname.row\column[x].property

Argument	Description
<i>gridname</i>	Name of the grid. Grid names cannot include spaces. When specifying the full <i>gridname</i> notation, you must also indicate the row, column, or cell.
<i>x</i>	Row or column, number, and property.

Example 1

Add all elements in row 3 of *mygrid1*.

`mygrid1.row[3].sum`

Example 2

Return the sum of the *nth* column:

`mygrid2.column[C].sum`

Example 3

Returns the absolute value of the cell located at the intersection of row 5 and column B in the current grid:

`ABS([5,B])`

Inserting a Formula Row or Column

► To insert a formula row or column:

- 1 In an existing grid, select **Insert**, then **Row** or **Column**, and then **Formula**.

The formula row or columns is inserted with a = # symbol.

- 2 In the formula bar at the top of the Report Designer, select a function from the drop-down list, and then enter the formula in the formula text box to the right of the drop-down.

See “Using the Formula Bar” on page 43.

Showing the Formula In a Row or Column Heading

► To show the formula in the heading for formula row or column:

- 1 In a grid, select the heading cell.
- 2 From **Heading Row Properties** or **Heading Column Properties**, select **Show Formula**.

Calculating at the Cell Level

You can create a custom cell-level formula on cells in a formula row or column. When you select a formula cell, you can select the following calculation preferences in **Cell Properties**:

- If the cell is an intersection of a formula row and formula column, you can choose whether the calculation is based on the row formula, the column formula, or a new custom formula.
- If a cell is located in a formula row, you can choose whether the calculation is based on a row formula or a new custom formula.
- If a cell is located in a formula column, you can choose whether the calculation is based on a column formula or a new custom formula.

[Table 1](#) shows an example of a cell that is calculated to display the % variance. Cell D4 displays the % variance between the sum of “Jan” and the sum of “Feb”; specifically, $((356-350) / 350) * 100 = 1.71\%$. A cell level formula gives the intended result by placing a custom formula in the cell D4. The formula needed to get the correct result is $(([B] - [A]) / [A]) * 100$.

Table 1 Reason for Using Cell Level Calculation

		A	B	C	D
	Total Corp	Jan	Feb	Var	%Var
1	Cola	100	105	5	5.00%
2	Root Beer	200	203	3	3.00%
3	Grape	50	48	(2)	(4.00%)
4	[1].sum	350	356	6	1.71%

► To calculate at the cell level:

- 1 Open a report and select a grid.
- 2 Select a formula cell or cells.
- 3 In **Cell Properties**, under **Calculation order**, select one of the following options:
 - **Use the row formula**—Applies the row calculation to the cells.
 - **Use the column formula**—Applies the column calculation to the cells.
 - **Use a custom formula**—Adds a new formula. (Enter the custom formula in the formula bar at the top of the Report Designer.)

Tip: You can select multiple contiguous formula cells in a formula row or column to change the calculation order.

Expanding Rows and Columns

Subtopics

- [Expressing the Segment Argument](#)
- [Expressing the Range Argument](#)
- [Identifying Multiple Segments with Ranges](#)

You can use the design-time row, column, and cell to reference the calculated rows, columns, or cells.

Row or Column Syntax:

AXIS[*Segment*(*Range*)] .*Property*

Cell Syntax:

CELL[*Row Segment*(*Range*), *Column Segment*(*Range*)] .*Property*

Parameters:

Parameter	Description
<i>AXIS</i>	Row or column containing the expanded cells identified by the row and column templates. For example, [3] refers to row 3 and [C] refers to column C. For example, [3] .sum in a column formula is calculated as the sum of row 3. The default is row.
<i>CELL</i>	Intersection of the expanded cells, given the row and column template coordinates, respectively. For example, [2,C] is the single cell where row 2 intersects column C.
<i>Segment</i>	Value that represents a row or column ID in the Report Designer. Referenced segments refer to design-time rows or columns, which can expand in the viewer. View-time rows and columns are the expanded rows and columns. Use numbers for row IDs and letters for column IDs. Note: If you are using an expanded formula row, the default segment is row. If you are using an expanded formula column, the default segment is column. For example, row [5] refers to expanded row 5. For more information on segments, see “Expressing the Segment Argument” on page 34 .
<i>Range</i>	Expanded cells used in the calculation of the function. Use colons to refer to a range of expanded rows or columns. For example, (3 : 6) addresses rows 3, 4, 5, and 6. For more information on ranges, see “Expressing the Range Argument” on page 35 .
<i>Property</i>	Function applied to aggregate rows, columns, or cells. For more information on using properties, See “Aggregate Property Arguments” on page 88 .

Expressing the Segment Argument

The Segment argument is a value that represents a row or column ID in the designer. Referenced segments refer to design-time rows or columns, which might expand in the viewer. View-time rows and columns are the expanded rows and columns. You can use numbers for row IDs and letters for column IDs. You can refer to one or more segments in any standard formula..

Table 2 Notation Used for Multiple Segments in Standard Formulas

Type	Notation	Description	Example
Single segment	[]	Single (design-time) segment	Point to segment 2: <code>row[2]</code> Note: Previous releases support this notation.
Nonconsecutive segment	[;]	Multiple single rows or column segments	Point to segment rows 2, 5, 7, 8 and 10: <code>row[2;5;7;8;10]</code> Note: Use a semicolon rather than a comma to delineate row or column references.
Segment Range	[:]	Range of rows or column segments	Point to segment rows 2, 3, 4 and 5: <code>row[2:5]</code>

The following examples show different combinations of the segment notation in [Table 2](#).

Example 1:

Point to segment rows 1, 3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15 and 17:

```
row[1; 3; 5:10; 12:15; 17]
```

Example 2:

If the following expression references row segment 20, but the grid contains only 15 row segments, the formula works only on segment 4:

```
row[4; 20]
```

If you enter a larger starting segment than ending segment, the system internally switches them around.

Example 3:

Show how the segments are internally switched:

```
row[6:3] becomes row[3:6]
```

Multiple references to the same segments are valid. This causes the same segments to be included twice in an expression.

Example 4:

Sum row 5 twice:

```
row[2;3;5;7;5].sum
```

Example 5:

Reference row segments 2, 3, 4, 5, 6, 7, and segment 5 again. Include segment 5 twice in the calculation of the average:

```
row[2:7;5].avg
```

Text segments are ignored.

Example 6:

Ignore row segment 3 if it is a text row:

```
row[1:5]
```

Expressing the Range Argument

The Range argument refers to view-time expanded rows and columns, on which calculations are performed after suppression. Sorting is performed after calculation is completed.

Table 3 Range Notation

Type	Notation	Description	Example
None		The Range argument is optional.	Point to all expanded rows in row segment 2. <code>row[2]</code>
Single Range	[segment (range)]	Single expanded row or column	Point to expanded row 5 in segment 2: <code>row[2(5)]</code>
Consecutive Range Reference	[:]	Range of expanded rows or columns	Point to expanded rows 5,6,7,8,9,10 in row segment 2: <code>row[2(5:10)]</code>
Nonconsecutive Range references	[,]	Multiple single expanded rows or columns	Point to expanded rows 5, 7, 8 and 10 in row segment 2: <code>row[2(5,7,8,10)]</code>

If a range is used, you can use any combination of the range notations in [Table 3](#).

Example 1:

Point to expanded rows 5 through 10 in design-time row 2:

```
row[2(5:10)]
```

If row segment 2 expands to 15 rows, the function operates on only expanded rows 5 through 10.

Example 2:

Identify columns C through E in design-time column A:

```
column[A(C:E)]
```

Example 3:

Point to expanded rows 1, 3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15 and 17 in design-time row 2:

```
row[2(1,3, 5:10, 12:15, 17)]
```

When the grid contains suppression properties, expanded rows and columns may be suppressed. If the Range argument refers to a number of expanded rows that is greater than the maximum number of rows expanded in the segment, the extra rows are ignored.

Example 4:

In the following example, expanded rows 6 through 10 are identified, but segment 1 contains only 3 expanded rows. In this situation, the function works only on expanded rows 6 through 8:

```
row[1(6:10)]
```

If none of the expanded rows in the Range exist, the entire axis reference is ignored.

Example 5:

Ignore the reference to 23 if segment 3 only contains 10 expanded rows:

```
row[3(23)]
```

Example 6:

If you enter a larger starting range than ending range, the system internally switches them around. The following shows how the range is reversed:

```
row[1(7:5)] becomes row[1(5:7)]
```

Multiple references to the same expanded rows or columns are valid. However, this may cause the same rows or columns to be included twice in an expression.

Example 7:

Sum expanded row 5 twice:

```
row[2(3, 5, 7, 5)].sum
```

Example 8:

Specify expanded rows 3, 4, 5, 6, 7, and row 5. Include row 5 twice in the calculation of the average:

```
row[2(3:7, 5)].ave
```

Identifying Multiple Segments with Ranges

The syntax for a range in a segment range is:

```
AXIS[Segment Start : Segment End (Range Start : Range End)]
```

where *Segment Start* and *Segment End* refer to multiple segments.

Example 1:

Point to expanded rows 5, 6, and 7 in row segments 1, 2, 3, and 4:

```
row[1:4(5:7)]
```

Example 2:

Point to expanded rows 5, 7, and 9 in row segments 1, 2, 3, and 4:

```
row[1:4(5,7,9)]
```

If the segment argument refers to segments that do not exist in the axis, the segments are ignored. This includes suppressed rows or columns.

Example 3:

Point to expanded rows 4, 6, 7,8, 9, and 10 in row segment 1 and 4, 5, and 6 in row segment 3 and all expanded rows in row segment 7:

```
row[1(4, 6:10); 3(4:6); 7]
```

Formatting Grids

Subtopics

- [Expanding the View of a Grid](#)
- [Formatting Numbers in Grids](#)
- [Using Special Currency Formats](#)
- [Aligning the Text in a Grid](#)
- [Hiding Rows, Columns, or Grids](#)
- [Modifying the Font](#)
- [Creating Custom Colors](#)
- [Adding Borders and Shading to Cells in Grids](#)
- [Freezing Grid Headers](#)
- [Inheriting Cell Formatting](#)
- [Copying Cell Formatting](#)
- [Merging Several Cells Into One Cell](#)

Expanding the View of a Grid

You can expand the view of a grid to the size of the Report Designer workspace. This is helpful when the grid contains a lot of rows and columns. The expanded view helps you to locate cells more easily.

➤ To expand the view of a grid :

- 1 In the Report Designer, select the upper left cell of a grid to select the entire grid..
- 2 In **Grid Properties**, select **Allow Expand All**.

Formatting Numbers in Grids

When you format how numbers are displayed in a grid, you can define the formatting options for an entire grid or for individual cells, rows, or columns:

► To format numbers in a grid:

- 1 In the **Report Designer**, select one or more cells, or select the entire grid.
- 2 Select **Format**, and then **Cell**.
- 3 In the **Format Cells** dialog box, in the **Number** tab, define the following options:
 - **Positive Number Symbols**—Select a prefix or suffix for positive numbers.
 - **Negative Number Symbols**—Select a prefix or suffix for negative numbers. To display negative values in red, select **Display in Red**.
 - **Decimal Places**—Select a number from 0 to 9 to specify the number of decimal places to apply to a number.
 - **Separator Characters**—How to define separator characters:
 - **Defined by Preferences**—Use the symbols specified in the user **Preferences** dialog box in the Oracle Hyperion Financial Reporting repository. (In the repository, select **File**, then **Preferences**, and then **Financial Reporting**, and the **Financial Reporting Studio**).
 - **Defined in Report**—Define a thousands separator and a decimal separator for the current report.
 - **Scale Value By**—Number to multiply values by for scaling purposes.
You can enter a number, or you can select a value from the drop-down list. The default scaling option is 1, which does not scale values. If you enter .001, the value 2,000 is displayed as 2.
- 4 Click **Apply**, and then click **OK**.

Using Special Currency Formats

Some currencies use specialized formats for their currency symbols. To add a special currency symbol to reports, the font used for that symbol must be installed on your local machine and the server to which you connect.

► To add a special currency symbol to a report:

- 1 **Open report and select a grid.**
- 2 **Select Format**, and then **Cell**.
- 3 In the **Format Cell** dialog box, in the **Font** tab, select the font for the currency symbol.
 - “AbakuTLSymSans” for the Turkish lira symbol
 - “Rupee Floradian” for the Indian rupee symbol

These fonts must have previously been installed in the system where your browser is running and on your server in order to select them.

- 4 Click **Apply**.
- 5 In the **Number** tab, in the **Prefix** or **Suffix** fields, select **Rupee (India)** or **Lira (Turkey)** to add the symbol as a prefix or suffix for numeric values.
- 6 Click **Apply**, and then click **OK**.

Note: If fonts other than “Rupee Floradian” or “AbakuTlSymSans” are used, the country currencies are displayed with the letters “RS” and “TL” respectively.

Aligning the Text in a Grid

► To align the text in a grid:

- 1 Open a report and select a cell or cells in a grid.
- 2 Select **Format**, and then **Cells**.
- 3 In the **Format Cells** dialog box, in the **Alignment** tab, define the following options:
 - **Horizontal**—Horizontal alignment in the report.
 - **Vertical**—Vertical alignment in the report.
 - **Indent**—Number of spaces to indent the heading and the associated text, data, and formula cells from the left margin.
 - **Indent Increases for Each Generation By**—Amount by which the indent increases for each generation of a member.

This option is enabled only for member rows and member selections that use member selection functions (for example “Children(East)” or “Descendents (Market)”). If you have an individual selection (for example “East” or “West”), you cannot define a value.

The indentation is done on the member’s “generation”, which is a number that denotes its hierarchy position from left to right. A dimension parent is Generation 1, and its children are Generation 2.

To indent the row headings for each generation of a selected member:

- a. Confirm that you selected a row with the desired heading cell.
- b. Select the number of spaces to indent the text.

This number is added to the number defined in **Indent**. To specify a reverse indentation (from right to left), enter a negative number.

Tip: Select a member relationship that returns members on multiple levels; for example: Children of...(Inclusive), Descendants of..., Descendants of...(Inclusive), Parents of...(Inclusive), Ancestors of, Ancestors of...(Inclusive), and AllMembers.

- 4 Click **Apply**, and then **OK**.

Hiding Rows, Columns, or Grids

You can hide rows, columns, or an entire grid so that the values are not displayed when you print or view a report. For example, you might want to hide a grid if you include a chart based on the grid data and want to display only the chart on the report.

Note: The data and calculations in rows or columns are evaluated, regardless of the formatting that is applied to them.

► To hide a grid:

- 1 In the **Report Designer**, select a grid.
- 2 In **Grid Properties**, expand **Suppression**, and then select **Hide Grid**.

► To hide a row or column in a grid:

- 1 In the **Report Designer**, select a row or a column in a grid.
- 2 In **Row Properties** or **Column Properties**, select **Hide Always**.
- 3 Specify whether ignore the values or calculations in the hidden row or column when you print or view a report.
 - To ignore values and calculations, select **Suppression Ignores Row** or **Suppression Ignores Column**.
 - To include values and calculations, clear **Suppression Ignores Row** or **Suppression Ignores Column**.

Modifying the Font

► To the font displayed in a grid:

- 1 Open a report and select a cell or cells in a grid.
- 2 Select **Format**, and then **Cell**.
- 3 In the **Format Cells** dialog box, select the **Font** tab and define the desired font option.
- 4 Click **Apply**, and then **OK**.

Creating Custom Colors

Financial Reporting Web Studio provides a palette of basic colors for text in reports and other artifacts. You can add to that palette by defining custom colors.

- To define a custom color:
 - 1 Open a report and select a cell or cells in a grid.
 - 2 Select **Format**, and then **Cells**.
 - 3 In the **Format Cells** dialog box, in the **Font** tab, click the arrow next to **Color**, and then click **Custom Color**.
 - 4 In the **Select Custom Colors** dialog box, define the custom color, and then click **OK**.

Note: Custom colors are not saved between Financial Reporting Web Studio sessions.

Adding Borders and Shading to Cells in Grids

You can apply borders and shading to add emphasis to selected cells, rows, or columns in a the grids. You can specify a border style and then apply it to any sides of the selected area. You can also select a shading color to apply shading to your selection.

- To add borders or shading to cells in a grid:
 - 1 Open report and select the cells to format in a grid.
 - 2 Select **Format**, and then **Cell**.
 - 3 In **Format Cells** dialog box, select the **Borders & Shading** tab.
 - 4 To specify a border, select a line style for the border under **Style**, and then click one or more of the border placement buttons.
 - 5 To add shading, select a color from the drop-down list under **Shading**.
 - 6 Click **Apply** to preview your selections, and then click **OK**.

Freezing Grid Headers

- To freeze the row and column headings of a grid:
 - 1 In the Report Designer, select the top left cell of the grid to select the entire grid.
 - 2 In **Grid Properties**, select **Freeze Grid Headers**.

Inheriting Cell Formatting

When formatting cells in a grid, you can inherit formatting from a cell and apply it to another cell. The following formatting options can be inherited from a cell in a grid:

- Number
- Alignment
- Font
- Borders and Shading

- Replacement Text


► To inherit cell formatting:

- 1 Open a report and select a cell or cells in a grid.
- 2 Select **Format**, and then **Cell**.
- 3 In the **Format Cells** dialog box, in the **Inherit Formatting** tab:

- Select the formatting options to inherit.

To select all the formatting options under a category, select the check box next to the category. For example, to inherit all number formatting options, select the check box next to Number.

To select specific formatting options for a category, click ►, and then select the specific formatting options. For example, to inherit the Decimal Separator value under Number, expand Number, and then select the check box next to Decimal Separator.

To inherit all the formatting options, click 

- Select the rows or columns from which to inherit the formatting.

You can select the Current Row or Current Column, or you can select a specific row or column number.

- 4 Click **Apply** and then **OK**.

Note: You can only inherit from a cell that does not have previously defined inheritance. For example, if you format column A and then inherit formatting into column B, you cannot then inherit formatting from column B into column C. You can, however, format column A, select columns B and C, and inherit formatting for column B and C from column A.


Note: If you have inherited formatting and then modify the format of the cell to which you inherited formatting, the inheritance you established no longer applies. For example, if you inherit the font name property in cell A4 from cell B4 and you subsequently modify the font name in cell A4, the format inheritance no longer exists. This breaks the formatting link that was previously established between the two cells.

Copying Cell Formatting

You can copy cell formatting from a cell or range of cells and apply it to a destination cell or range of cells in a grid.


► To copy cell formatting:

- 1 In the Report Designer, select or create a grid.
- 2 Format a cell or a range of cells.

- 3 Select the cell or range of cells containing the formatting to copy, and then click .
- 4 Select the destination cell or range of cells to apply the copied formatting.

The formatting is changed when you release the mouse.

Note:

Using  to copy cell formatting does not copy inherited formatting or conditional formatting.

Merging Several Cells Into One Cell

You can combine two or more cells into one cell in a grid. For example, you can combine several cells to create one data cell, text cell, or formula cell.

When you merge cells, the value and format of the upper-left cell is placed into the resulting merged cell. When the merged cell is separated into individual cells, all resulting cells inherit the formatting of the previously merged cell.

- To merge individual cells into a single cell, right-click the cells to merge, and then select **Merge**.
- To separate a merged cell back into individual cells, right-click the cells to merge, and then select **Merge**.

Using the Formula Bar

Subtopics

- [Creating Formulas](#)
- [Deleting a Formula Row or Column](#)
- [Modifying Member Selection Using the Formula Bar](#)
- [Example Report with Formulas](#)

The formula bar provides a central location to create formulas and select members when setting up grids in Financial Reporting Web Studio. The formula bar contains a text box to enter formulas and a drop-down where you can select functions when building formulas. If you select a member in a grid, member selection is activated on the formula bar.

The formula bar is displayed below the toolbar so you can easily access its functionality. It is visible but disabled when you are not using grids.

Creating Formulas

When you select a cell, column, or row containing a formula, you can apply a unique formula to that cell. When you create formulas, keep in mind the following points:

- Use row or column formulas as opposed to cell formulas whenever possible.

- Use reference properties instead of functions when possible.

Rows, columns, or cell references can have a property associated with them. For example, the row reference: `[1 : 5] . sum` produces the same result as using the Sum function; `Sum ([1 : 5])`. However, the first formula executes faster since it is being used as a reference property.

- Avoid using cross-axis references and consider using a cell reference if possible.
- Use parentheses only when necessary.

Improper use of parentheses can result in unnecessary evaluation iterations within the evaluation routine of the calculation engine.

► To create a formula:

- 1 In a grid, insert a formula row or column and select the formula cell.

To apply a formula to the entire column or row, select the column or row.

When you use a cell formula in a row or column formula, the cell formula should reference the cells exactly (intersection) and not just the row and column.

See “[Defining Formula Rows and Columns](#)” on page 30.

- 2 Select a function from the drop-down list, and then enter the formula in the formula text box to the right of the drop down.

See [Chapter 10, “Using Functions.”](#)

- 3 Click  to validate the formula.

Deleting a Formula Row or Column

► To delete a formula row or column:

- 1 Select the formula row or column.

- 2 Click  next to the formula in the formula bar.

Modifying Member Selection Using the Formula Bar

When working with members in a grid, you can use the formula bar to modify the selected members.

► To modify member selection using the formula bar:

- 1 In a grid, select a heading cell of a data row or column.
- 2 Click the button for the dimension next to the formula bar.
- 3 In the **Select Members** dialog box, select or remove members.

The new members are displayed in the grid. For information on member selections, see [“Assigning Members Using Functions” on page 68.](#)

- 4 Click  to accept the member entry.

The grid is set to obtain data for the new member/member function.

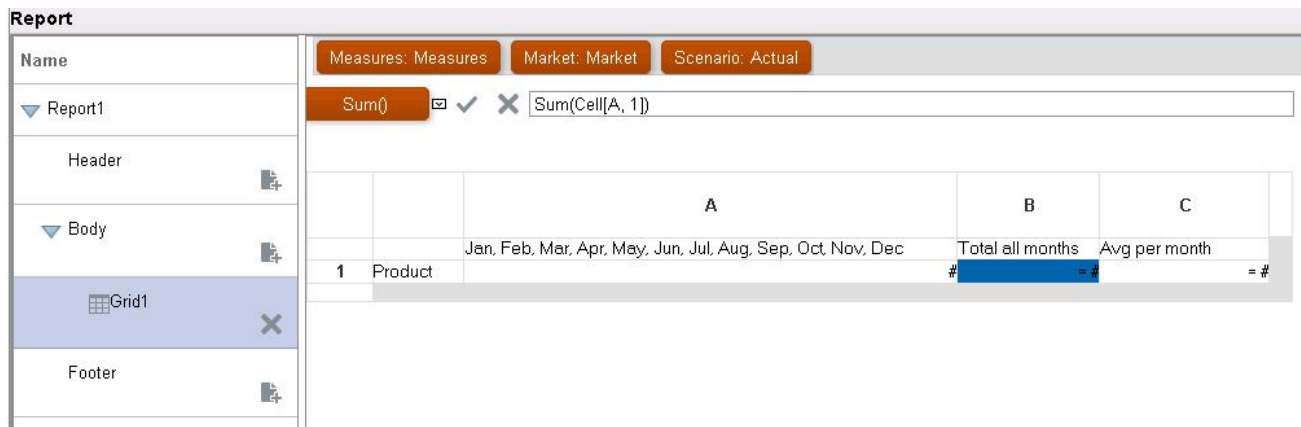
When modifying member selection using the formula bar, keep in mind the following points:

- Member names in the formula bar is not case-sensitive.
- A member function can be entered dynamically on data heading cells. See [“Assigning Members Using Functions” on page 68.](#)
- You cannot combine numerical functions with member functions; for example, `Sum(Children of (Product) (Inclusive))`.

Example Report with Formulas

Figure 1 shows the use of a formulas to summarize data for the year.

Figure 1 Formula That Summarizes the Data For the Year



The screenshot shows a report editor interface. On the left is a tree view with 'Report1' expanded to show 'Header', 'Body', 'Grid1', and 'Footer'. The main area displays a report grid. Above the grid, there are filters for 'Measures: Measures', 'Market: Market', and 'Scenario: Actual'. A formula bar shows the formula `Sum(Cell[A, 1])`. The grid has columns for months (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec) and summary rows for 'Total all months' and 'Avg per month'. The 'Product' dimension is selected for the row.

		A											B	C	
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total all months	Avg per month
1	Product													#	= #

The report resulting from this formula will include:

- Monthly total figures for the “Product” dimension
- A total for all months calculated using the `Sum` function
- An average amount per month, calculated using the `Avg` function

To create this example using Financial Reporting Web Studio:

1. Create a grid with “Product” for the row and “Year” for the column.
2. Double-click “Year” to bring up member selection.
3. Place all twelve months in the selected pane and remove “Year.”

4. Insert two formula columns: one for the “annual total ” and another for the “average monthly amount.”

In this example:

- The first formula column that falls outside the months of the “Year” member contains a `Sum` function that adds the figures for each month. Since all months are defined in one cell, the reference is to that cell location.

The formula is:

```
Sum(Cell [A1])
```

- The second formula column calculates the average of the months for the “Year” member. Because there are 12 months in a year, the `Avg` function adds all monthly totals and divides the total by 12.

The formula is:

```
Average(Cell [A, 1])
```

In general, there are two ways to specify a data formula in a grid, a row or column formula, or a cell formula:

- **Row/Column**—Apply a formula to the entire row or column, relative to each cell in that row or column.

To define the formula, select the formula row or column header to highlight the entire row or column, and then enter the formula in the formula bar. There are some performance benefits when applying row/column formulas.

- **Cell**—Apply a formula only to cells in a formula row or column.

To define the formula, select the cell, then select **Use a Custom Formula** in the Cell Properties, and then enter the formula in the formula bar.

If the cell intersects a formula row and formula column, you can select to use the row formula or the column formula as the cell formula. For more information, see [“Calculating at the Cell Level” on page 32](#).

If a formula repeats for each cell in a row or column, use a “Row/Column” formula. If different formulas are performed on each cell, use a “cell” formula.

Setting Data Query Optimization Options

When working with grids, you can set data query optimization options to optimize the efficiency of your data queries. Because data queries extract values from a multidimensional database connection, it is beneficial to speed up the data query process using the optimization settings.

► To set data query optimization options:

- 1 In a report that contains a grid, do one of the following:

- Select **Task**, and then **Data Query Optimization Settings**.

- Click the top left cell in the grid to select the grid, and then right-click and select **Data Query Optimization Settings**.
- 2 Select whether to suppress rows on the **Database Connection Server** or the **Report Server**.
 - 3 **Optional:** Click **Set to Default** to use the settings defined here as the default data query optimization settings.
 - 4 Click **OK** to save the settings.

Showing Supporting Detail for Planning

When you use a Planning database connection and select a row, you can display supporting detail for a member. If supporting detail exists for one or more cells in the selected row, additional rows are inserted to show the supporting detail. In addition, you can control whether a page break is allowed within the rows of supporting detail.

- To show supporting detail for a Planning database connection:
 - 1 In a grid, select a row that contains a member with supporting detail.
 - 2 In **Row Properties**, select **Display Supporting Details**.
 - 3 Specify whether to allow page breaks in the rows of supporting detail:
 - To allow a page break in the rows of supporting detail, select **Allow Page Break Within**.
 - To keep the parent member and the supporting detail on the same page, clear **Allow Page Break Within**. If the supporting detail does not fit on the current page, the parent member and supporting detail is moved to the next page.

If the supporting detail spans multiple pages, this option defaults to **Allow Page Break Within** and page breaks are used.

- 4 **Optional.** To position the supporting detail before or after the parent member, in the **Grid Properties**, under **Position**, select an option in **Position of Supporting Detail**.
- 5 **Optional.** To indent each generation in the supporting detail:
 - a. Select the heading cell of the member that is going to display the supporting detail.
 - b. Select **Format**, then **Cells**, and then the **Alignment** tab.
 - c. In **Indent Increases for Each Generation by**, select a positive number to indent (to the right) each generation in the supporting detail list, or select a negative number to indent in reverse (to the left) each generation in the supporting detail list.

This property applies only to the heading cells in the rows. It is enabled for Planning database connection only when **Display Supporting Detail** is selected.

- 6 **Optional.** To apply additional formatting to supporting detail, use conditional formatting. The following two options are useful:
 - **Supporting Detail**—If, at runtime, supporting detail results from a cell, you can format the supporting detail.

- **Relative Generation**—You can format the different generations of supporting detail independently. For example, if the resulting hierarchy of a member includes four generations of supporting detail, you can specify to format the third generation. The conditions would resemble the following example:

```
Condition 1: If  
Supporting Detail Is True    And  
Relative Generation (of) Market = 3
```

Note: In order for all types of database connections to use relative generation, you must select a member relationship that returns members on multiple levels. For example, Children of...(Inclusive), Descendants of..., Descendants of...(Inclusive), Parents of...(Inclusive), Ancestors of, Ancestors of...(Inclusive), and AllMembers. For Planning, you can use relative generation with a member or dynamic member if supporting detail is turned on.

3

Working With Text Objects

In This Chapter

Adding Text Objects.....	49
Adding a New Text Object.....	49
Inserting an Existing Text Object.....	50
Formatting Text Objects.....	50

Adding Text Objects



Text objects are report objects that can contain text or text functions. For example, you can enter a label, description, or a function that retrieves the current date in a text box. You can enter multiple paragraphs of text. Like other report objects, you can move, resize, or reposition the area containing the text.

You can create a new text object or insert an existing text object into the report. New text objects are empty areas where you can enter text, text functions, or rich text from an editor or word processor. Text objects are stored in the repository and can be inserted into numerous reports. For example, you can add the same text object in all your report designs.

Adding a New Text Object

► To add a new text object:

1 Do one of the following:

- Click , and then position the text object in the Report Designer.
- Select **Insert**, then **Text**, and then position the text object in the Report Designer.
- In the **Header**, **Body**, or **Footer** section of the Report Designer, click , and then select **Text**.

2 Enter the text in the text box.

To format the text, use the tool bar at the top of the text object. You can modify the properties of a text object under **Text Properties**. To insert a function in the text object, click **Insert Function** under **Text Properties**.

Inserting an Existing Text Object

- To insert an existing text object from the repository:
 - 1 Open a report and select **Insert**, and then **Saved Object**.
 - 2 In the **Type** drop-down list, select **Text**.
 - 3 Select the directory where you want to search for the text object, then select the desired text object.
 - 4 Perform an action:
 - To use the text object as a linked object, select **Link to Source Object**.
 - To use the text object as an unlinked object, deselect **Link to Source Object**. This is the default option.
 - 5 Click **Insert**.

Formatting Text Objects

Subtopics

- [Defining Fonts for Text](#)
- [Adding Borders and Shading to Text](#)
- [Aligning Text](#)

Defining Fonts for Text

- To define the font for text:
 - 1 Open a report and select the text in a text box.
 - 2 Define the font using the formatting options on the format bar above the text box.

Adding Borders and Shading to Text

- To add border and shading to text:
 - 1 Open a report and select the text in a text box.
 - 2 In **Text Properties**, select **Show Border**, and then select an option from the **Shading** drop-down list.

Aligning Text

- To align the text in a report:
 - 1 Open a report select a text box.

2 In **Text Properties** under **Position**, select a **Horizontal** and a **Vertical** position.

4

Working With Images

In This Chapter

Inserting Images	53
Formatting Images	54

Inserting Images

You can add images to the header, footer, or body of a report. After you add an image to a report, you can format the image. The following image formats can be inserted into a report in Financial Reporting Web Studio:



- Bitmaps: `.bmp`
- Graphics Interchange Format (GIF): `.gif`
- Joint Photographic Experts Group (JPEG): `.jpg`

You can insert an image file that is stored in a file system outside Financial Reporting Web Studio into the image frame, or you can insert an entire image object that is saved in Financial Reporting Web Studio.

Note: Image are saved with the report. As a result, the speed of the application may be affected based on the size of the image.

➤ To insert an image located outside of Financial Reporting Web Studio:


1 Do one of the following:

- Click , and then draw or position the image in the Report Designer.
- Select **Insert**, then **Image**, and then draw or position the image in the Report Designer.
- In the **Header**, **Body**, or **Footer** section of the Report Designer, click , and then select **Image**.

2 In **Image Properties**, click **Browse**, and then select the image to insert.

➤ To insert an image from the repository:

1 Do one of the following:

- Open report and select **Insert**, and then **Saved Object**.
 - In the **Header**, **Body**, or **Footer** section of the Report Designer, click , and then select **Saved Object**.
- 2 In the **Insert Saved Object** dialog box, in the **Type** drop-down, select **Image**.
 - 3 Select the desired image, and then click **Insert**.
 - 4 Click **Insert**.

Formatting Images

You can format the display of images in a report. For example, once you place an image, you can stretch the image, insert a page break before the image, and specify horizontal and vertical positions for the image.

► To format an image in a report:

- 1 Open a report and select an image.
- 2 In **Image Properties**, define the following information:
 - **Page Break Before**—Places a page break before the image.
 - **Stretch**—Expands or shrinks the image to fill the size of the image boundary in the Report Designer workspace.
 - **Position**—Options to position the image in the report.

Tip: To move the image, select the image, and with all borders activated, drag it to another location. Do not click and drag the borders of the image, because this stretches the image vertically or horizontally. Be sure to select the entire image.

5

Working With Charts

In This Chapter

About Charts.....	55
Inserting a Chart	56
Defining Chart Data.....	56
Defining Chart Titles, Legends, and Labels	57
Formatting Charts	60
Positioning Charts in Reports	63
Resizing and Moving Charts.....	63

About Charts

A chart is a graphical representation of report data from a grid on a report. The chart retrieves data dynamically from the grid, so if data in the grid changes, the chart is updated automatically. You create charts to portray data graphically in your reports. You can use the different chart types to illustrate trends and tendencies or to highlight differences and improvements.

Types of Charts

- Bar Chart
- Line Chart
- Pie Chart
- Combo ChartA combo chart is a combination bar chart and line chart. Combo charts are typically used for comparison charting. The combo chart accepts a group of bar data sets and line data sets. It uses all of the display parameters found in bar charts and line charts. Two data set selections are required, one for bar rendering and one for line rendering.



Chart Components

- Chart type and data content
- Chart attributes such as title, title angle, font angle, and background formatting
- Legend and axis information such as labels, font, background color, and orientation
- Formatting options specific to a chart type, such as bar width for a bar chart
- Horizontal and vertical positioning of a chart on a report

Inserting a Chart

► To insert a chart:

1 Do one of the following:

- Click , and then position the chart in the **Body** section in the Report Designer.
- Select **Insert**, then **Chart**, and then position the chart in the **Body** section in the Report Designer.
- In the **Body** section in the Report Designer, click , and then select **Chart**.

2 Modify the chart using the options in **Chart Properties**.

► To insert a chart object from the repository:

1 Open a report and select **Insert**, and then **Saved Object**.

2 In the **Type** drop-down list, select **Chart**.

3 Select the desired chart object, and then do one of the following:

- To use the chart object as a linked object, select **Link to Source Object**.
- To use the chart object as an unlinked object, clear **Link to Source Object**. This is the default.

4 Click **Insert**.

Defining Chart Data

When defining chart data, you identify the rows and columns from the controlling grid that contain the data to display. You can reference data or formula rows and columns by selecting contiguous or non-contiguous rows and columns for the data range. You can also include or exclude auto calculations from the data set.

If you are designing a line, bar, or combo chart, you can reference multiple rows or columns from the grid. If you are designing a pie chart, you can reference any single row or column from the grid. When you define a combo chart, you specify the data rows and columns for the bar and line.

Charts display aggregate rows or aggregate columns that expand. For example, if you specify row 1 and row 1 contains a function that retrieves ten child members, the chart displays ten data sets, or one data set for each child member.

► To define chart data:

1 Select the chart in the Report Designer.

2 In **Chart Properties**, under **Grid**, select the grid that the chart will reference.

The grid must exist in the current report.

- 3 In **Legend Items from Grid**, select **Rows** to reference row data or **Cols** to reference column data.
- 4 In **Data Range**, under **Rows** and **Cols**, select the rows and columns to include in the chart.
When selecting a data range, you can specify contiguous or non-contiguous rows and columns.
- 5 **Optional**. Select **Include Auto Calculation** to include auto calculations from the data set in the chart.

Defining Chart Titles, Legends, and Labels

Subtopics

- [Defining Chart Titles](#)
- [Defining Chart Legends](#)
- [Defining Chart Axis Titles](#)
- [Defining Axis Labels](#)

You can format chart titles, chart legend titles, chart axes titles, and axis labels in a chart. You can also format the font and borders and background of the titles and axes you indicate..

Defining Chart Titles

► To define a chart title:

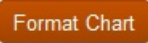
- 1 Open a report and select a chart.
- 2 In **Chart Properties**, click .
- 3 In the **Appearance** tab, in the **Chart Title** text box, enter a title for the chart.



To print the title on multiple lines, use \n in the title. To indicate a possessive apostrophe, use \'s. You can combine these commands to use apostrophes with titles that span multiple lines.

- 4 Enter other desired formatting options on the **Appearance** tab, and then click **OK**.

Defining Chart Legends

► To define chart legend options:

- 1 Open a report and select a chart.
- 2 In **Chart Properties**, click .
- 3 In the **Legend** tab, define the following options:
 - Select **Place Legend** to select the vertical and horizontal position of the chart legend relative to the chart boundaries.

- Click  to format the border and background of the legend.
- Click  to format the font used in the legend title and its labels.
- In **Legend Title**, enter a title for the legend.
- In **Suppress Repeating Labels**, select an option:
 - **Inherit Settings from Grid**—Inherit the settings from the grid to which the chart is attached.
 - **On**—Suppress x-axis labels that would otherwise repeat.
 - **Off**—Do not suppress x-axis labels that would otherwise repeat.

4 Click **OK**.

Defining Chart Axis Titles

Metadata titles and primary and secondary axis titles identify the data that is displayed on the axes of a chart.

Note: Chart axis options are available for bar, line, and combo charts only.



► To define a chart axis title:

1 Open a report and select a chart.

2 In **Chart Properties**, click .

3 In the **Axes** tab, define the following options:

- In **Metadata Title**, enter the title for the horizontal axis.
- In **Primary Axis Title**, enter the title for the primary vertical axis. This is displayed to the left of the chart.
- In **Secondary Axis Title**, enter the title for the secondary vertical axis. This is displayed to the right of the chart.

4 For any of the options, click  to format the border and background of the axis title, and click  to format the font.

5 Click **OK**.

Defining Axis Labels

You can define and format labels for the primary and secondary axis in a chart. For a combo chart, you can specify and format a secondary axis label. You can also display and format metadata labels.

Note: A primary or secondary axis label enables you to specify a custom range of values on the axis.

► To define axis labels:

1 Open a report and select a chart.

2 In **Chart Properties**, click .

3 In the **Axes** tab, define the following options:

- Select **Show Metadata Labels** to show metadata labels with a standard display.

Financial Reporting implements staggered labels as a default to ensure the even distribution of data on report axes.

- Select **Primary Axis Label** to use a range of values on the primary axis.

You can use the default range of values or a custom range of values on the primary axis label.

- To use the default range of values, select **Primary Axis Label**.
- To use a custom range of values, select **Primary Axis Label**, and then enter a range of values, including minimum, maximum, and step.

You can use a custom range to change the starting and ending numbers of your axis and the numeric intervals. For example, you can display values 40 through 60 at three step intervals by entering 40, 60, and 3 as the values **Min**, **Max**, and **Step**.

- Select **Secondary Axis Label** to use a range of value on the secondary axis.

You can use the default range of values or a custom range of values on the secondary axis label.

- To use the default range of values, select **Secondary Axis Label**.
- To use a custom range of values, select **Secondary Axis Label**, and then enter a range of values, including minimum, maximum, and step.

- To enter a range of values, including minimum, maximum, and step, on the secondary axis in a bar or line chart, select the check box for **Secondary Axis Label** and the **Custom Range** check box.

4 For any of the options in the **Axes** tab, click  to format the border and background, and click  to format the font.

5 Click **OK**.

Formatting Charts

Subtopics

- [Formatting Fonts](#)
- [Formatting Borders](#)
- [Formatting Bars in Bar and Combo Charts](#)
- [Formatting Lines in Line and Combo Charts](#)
- [Formatting Combo Charts](#)
- [Formatting Pie Charts](#)
- [Formatting Background and Gridline Options](#)
- [Defining Styles for Data Sets](#)

You format charts to customize their appearance and layout in your reports. You can also format charts to portray data using different chart types.



► To format a chart:

- 1 Select a chart in the Report Designer.
- 2 In **Chart Properties**, click .
- 3 In the **Format Chart** dialog box, define the formatting options for the chart.

The tabs that are displayed in the **Format Chart** dialog box depend on the type of chart you are formatting.

Formatting Fonts


► To format the fonts used in a chart:


- 1 Open a report and select a chart.
- 2 In **Chart Properties**, click .
- 3 In the tabs in the **Chart Properties** dialog box, click .
- 4 In the **Fonts** dialog box, format the fonts, and then click **OK**.

The **Fonts** dialog box enables you to preview your font option selections before returning to the chart.

Formatting Borders

► To format the chart borders:

- 1 Open a report and select a chart.
- 2 In **Chart Properties**, click .

- 3 In the tabs in the **Chart Properties** dialog box, click .
- 4 In the **Borders** dialog box, format the borders, and then click **OK**.

Formatting Bars in Bar and Combo Charts

You can display bars in a bar or combo chart by grouping data sets and specifying their orientation on the chart. You can also display point values in a bar or combo chart. You can modify the shapes and widths of bars and bar borders.

Use the following guidelines to specify how bars are displayed in your charts:

- Vertical orientation starts the bars on the horizontal axis and displays the bars vertically.
- Horizontal orientation starts the bars on the vertical axis and displays the bars horizontally.
- The bar width represents the relative width of the bar as a percentage of available space. A smaller number results in more space between bars.
- The bar shape represents the shape of the displayed bars.
- You can apply shading (pattern and color) by data set using the Element Style tab of the Format Chart dialog box.

► To format the bars in bar and combo charts:


- 1 Open a report and select a bar or combo chart.
- 2 In **Chart Properties**, click .
- 3 In the **Bar Options** tab, define the formatting options, and then click **OK**.

Note: Bar charts may not display point values if the data values do not fit in the bar. If point values do not display, re-size the chart or apply scaling to the data values in the grid to make the numbers smaller.

Formatting Lines in Line and Combo Charts

You can specify display options for lines in line and combo charts. For example, you can display lines in separate rows, stacked as a percentage of the total of all values, or stacked as absolute values.

► To format lines in line and combo charts:

- 1 Open a report and select a line or combo chart.
- 2 In **Chart Properties**, click .
- 3 In the **Line Options** tab, define the line type, and then click **OK**
 - **Rows**—Displays the line sets in separate rows. For example, if you define the 3D depth as nonzero, the lines are displayed in separate rows, from front to back.

- **Percent**—Displays the line sets stacked on top of each other, normalized to 100%. As each line set is drawn, its values are added to previous values displayed and are shown as percentages of the total of all values.
- **Stack**—Displays the line sets stacked. As each line set is drawn, its values are added to previous values

Note: You can define styles for the lines representing the current data set. See [“Defining Styles for Data Sets” on page 63](#).

Formatting Combo Charts

A combo chart combines a line chart and a bar chart. Typically used to contrast data sets, combo charts employ bar and line options and are set with the same variables as bar charts and line charts.

When you are formatting a combo chart, the Format Chart dialog box includes the Line Options tab and Bar Options tab. See [“Formatting Bars in Bar and Combo Charts” on page 61](#) and [“Formatting Lines in Line and Combo Charts” on page 61](#).

Formatting Pie Charts

When creating a pie chart, you can specify the starting angle for the first pie slice, the level of separation between pie slices, the position and type of pie slice labels, and the slice border style and width. You can also set font options including font, font size, effects, and angle for the textual display of the slice values.

➤ To format a pie chart:

- 1 Open a report and select a pie chart.
- 2 In **Chart Properties**, click .
- 3 In the **Pie Options** tab, define the formatting options, and then click **OK**.

Note: You can define styles for the slices of the pie that represent the current data set. See [“Defining Styles for Data Sets” on page 63](#)

Formatting Background and Gridline Options

You can set background options for a chart, such as a border, gridlines, gridline color, gridline style, and grid background. You can also set the grid depth.

➤ To format background and gridline options:


- 1 Open a report and select a chart.

- 2 In **Chart Properties**, click .
- 3 In the **Appearance** tab, select options to define background and gridline options, and then click **OK**.

Defining Styles for Data Sets

You can define a pattern and color for each data set in your report. The specified shading and patterns are applied to each data set to differentiate values in the chart.

► To define styles for a data set:

- 1 Open a report and select a chart.
- 2 In **Chart Properties**, click .
- 3 In the **Element Style** tab, select a data set, and then define theme and colors.
- 4 Click **OK**.

Positioning Charts in Reports

You can specify a chart's position on the report and how it is displayed when printed. You can specify a position relative to the grid, a position relative to other report objects, or no position at all. If you do not specify a position, the chart remains in the location where it was originally created in the report.

► To position a chart in a report:

- 1 Open a report and select a chart.
- 2 In **Chart Properties**, under **Position**, define the horizontal and vertical position of the chart.

Resizing and Moving Charts

You can change the size of a chart and move a chart to another location in a report.

► To resize a chart:

- 1 Click the name of the report that contains the chart to return to the report "layout view".
- 2 Click the chart.
The chart's border is highlighted in red to indicate that the chart is selected.
- 3 Click and drag a border on the perimeter of the chart to change the size.

► To move a chart to another location in the report:

- 1 Click the name of the report that contains the chart to return to the report "layout view".
- 2 Click the chart.

The chart's border is highlighted in red to indicate that the chart is selected.

3 Hold down the left mouse button and drag the chart to another location in the report.

Tip: You must be in the “layout view” to move or resize a chart. If you click on a chart in the **Body** section of the Report Designer, you cannot move or resize the chart.

6

Defining Members

In This Chapter

About Defining Members	65
Assigning Members	65
Searching For Members	70
Previewing Selected Members	71
Copying Members Names From a Text Application	71
Removing Members	71
Selecting Multiple Members	72
Access Privileges for Members in the Planning Database Source.....	73

About Defining Members

When you specify data for a grid, you set criteria that identifies the data to retrieve. You set the criteria by assigning members to the rows, columns, and page axes in the grid, as well as to the grid point of view and user point of view bars.

You can create lists of reusable members, and use functions to retrieve members dynamically. Before you define members, you must insert a grid in the report and define its dimension layout. (See [Chapter 2](#), “Working With Grids.”)

Assigning Members

Subtopics

- [Assigning Members to Data Rows or Columns](#)
- [Assigning Members to the Page Axis](#)
- [Defining Special Types of Members](#)
- [Assigning Members Using Functions](#)

You assign members and member lists to retrieve data for your reports. You can select members to be displayed in your report, or you can use functions to select dimension members dynamically. You can insert a separate row or column in a grid for each selected member, or you can place all of the selected members in one row or column.

Assigning Members to Data Rows or Columns

► To assign members to data rows or columns:

- 1 **Open a report and select a grid.**
- 2 **Open the **Select Members** dialog box by doing one of the following:**
 - Double-click a dimension in the grid.
 - Select a dimension in the grid, and then click the button in the formula bar with the dimension name.
- 3 **In the **Select Members** dialog box, select the members, members lists, or functions to assign to the selected data row or column.**
- 4 **Optional: To insert a row or column for each selected member, select **Place selections into separate rows (or Columns)**.**
- 5 **Click **OK**.**

Assigning Members to the Page Axis

You can assign members to the page axis of a grid. When you select members for the page axis and run the report, a grid is generated for each page member. If a chart references a grid with multiple page members, a new chart is generated for each page member.

For example, assume that you assign the following members in the Scenario and Entity dimensions to the page axis:

- Target 1 and Target 2 in the Scenario dimension
- Radio Promotions and Web Promotions in the Entity dimension

When you view the report, it is formatted and populated with data from all combinations of members on the page axis. The report contains four grids, one for each page member combination.

In this example, you could select any combination of the members:

- Target 1: Radio Promotions
- Target 1: Web Promotions
- Target 2: Radio Promotions
- Target 2: Web Promotions

Note: In Financial Reporting, all the members and corresponding data in the page axis are retrieved when the report is run. As a result, performance may be impacted for reports with a large number of members on the page axis.


► To assign members to the page axis:

- 1 **Open a report and select a grid.**

2 In the **Pages** drop-down list, select a dimension.

3 On the formula bar, select the button with the dimension name.

If you know the member names for a dimension, enter the member name into the formula bar text area. For example, for a “Measures” dimension, you could enter “Profit, Sales”.

Click  to verify the member selection.

4 In the **Select Members** dialog box, select the members to use on the page axis, and then click **OK**.

Defining Special Types of Members

Subtopics

- [Setting Up Report Prompts](#)
- [Setting Up a Current Point of View](#)

You can use the Select Members dialog box to select the following types of members:

- **Prompt**—Acts like a variable that asks you to select members when you run a report. You determine the dimension and defaults at design time, and you select the members when you generate the report. Prompt also enables the report writer to filter the dimension members that are available to select when running the report.
- **Current Point of View**—Acts as a variable that employs the point of view to specify the member of a dimension when the report is run.
- **User Point of View**—Can be selected as a member in the book point of view. The parameters can be modified by the end user.

Setting Up Report Prompts

A prompt is a way of assigning members to a dimension. The prompt is set up during design time. As a report designer, you can limit the type of data to assign to a dimension. You can also specify whether the Respond to Prompts dialog box displays member names, descriptions or aliases, or both member names and descriptions or aliases.

For example, you could create a report that enables the user to select from lines of products. The report designer selects Prompt as a member of Product, and then selects specific product lines. The person who runs the report then selects from those product lines.

► To set up a report prompt for members:

1 Open a report and select a grid.

2 Double-click the dimension cell for which you want to assign a prompt.


3 In the **Select Members** dialog box, on the **Members** tab, select **Prompt for <dimension name>**, and then click  to select it.

4 Click **OK**.


5 In the **Define Prompts** dialog box, enter the following information:

- **Title**—Enter the title for the prompt.

For example, for a “Measures” dimension, you could say “Enter Measures”.

- **Default Member**—Enter the member to use as the default member, or click  to select the default member. If you specify multiple members as the default, separate the members by commas.

If you leave **Default Member** blank, the **Respond to Prompts** dialog box does not display a default member when you run the report or create a batch.


- **Choices List**—Click  to select the member or members available as choices when responding to the prompt. If you specify multiple members in the choices list, separate the members by commas.
- Select which labels to display in the **Respond to Prompts** dialog box when you run the report (**Member Name**, **Alias**, or **Member Name and Alias**).

6 Click **OK**.

Setting Up a Current Point of View

A current point of view enables you to obtain a member for a row, column, or page from the user point of view.

► To set up a current point of view:

- 1 Open a report and select a grid.
- 2 Double-click the dimension cell for which you want to set up a current point of view.
- 3 In the **Select Members** dialog box, on the **Members** tab, select **Current Point of View for <dimension name>**, and then click  to select it.
- 4 Click **OK**.

Assigning Members Using Functions


Subtopics

- [Functions Available Through Member Selection and the Formula Bar](#)
- [Functions Available Only Through Member Selection](#)



You can use functions to select members dynamically. After you select a function, you can edit its parameters.

► To assign members using functions:

- 1 Open a report and select a grid.

- 2 Double-click a dimension cell.
- 3 In the **Select Members** dialog box, select the **Functions** tab.
- 4 Highlight one or more functions, and then click  .

You cannot select functions that do not require parameter values, such as BottomofHierarchy, TopofHierarchy, AllMembers, and SuppressSharedMembers.

- 5 Select a function, and then click  .
- 6 Click  , and then select the members to add to the function.

Functions Available Through Member Selection and the Formula Bar

The functions listed in [Table 4](#) can be selected from the **Functions** tab in the **Select Member** dialog box, and they can be entered dynamically in the formula bar.

Table 4 Functions Available Through Member Selection and the Formula Bar

Function	Parameters Needed	Description
AllMembers	All Members in Hierarchy <i>dimension</i>	Members of the specified dimension
Ancestors	Ancestors of <i>member</i> (Inclusive)	Members at all levels above the specified member
BottomOfHierarchy	Bottom of Hierarchy <i>dimension</i>	All Level 0 members
TopOfHierarchy	Top of Hierarchy <i>dimension</i>	Top-level member of hierarchy
Children	Children of <i>member</i> (Inclusive)	Members one level below the specified parent
Descendants	Descendants of <i>member</i> (Inclusive)	Members at all levels below the specified parent
Parent	Parent of <i>member</i> (Inclusive)	Member one level above the specified member
Siblings	Siblings of <i>member</i> (Inclusive)	Members with the same parent as the specified member
Member	<i>member</i>	The specified member
Members	N/A	Members of the current dimension
OfSameGeneration	Same generation as <i>member</i>	Members of the same dimension and generation as the specified member
OnSameLevelAs	Same level as <i>member</i>	Members from the same dimension and on the same level as the specified member
SameLevelAs	N/A	Members of the same level
SystemMemberList	System-defined Member List Lev(n), <i>dimension</i>	Members in a specified system-defined member list

Functions Available Only Through Member Selection

The functions listed in [Table 5](#) can be selected from the **Functions** tab in the **Select Member** dialog box. They cannot be entered dynamically in the formula bar.


Table 5 Functions Available Only Through Member Selection

Function	Description
Match	Extracts members that match a specified pattern or set of characters. Note: When using the asterisk (*) wildcard, the pattern can contain only one asterisk and it can only appear as the last character.
OrderBy	Defines the order of members that come back from member selection by a member property. Required parameters: <ul style="list-style-type: none">● Property—Provides the Member Name option and a list of all properties available for members.● Direction—Provides the options for directions: Ascending, Descending, Hierarchy Ascending, Hierarchy Descending. OrderBy applies to all member selections in the Selected list.
Property	Extracts members with the specified property value.
Range	Defines a range of members. For example, you can use the “Year” dimension to specify all months in the first quarter by selecting “January” as the start member and “March” as the end member.

Searching For Members

You can perform searches for members to edit. You can search using a text string or by property (that is, name and description). You can also use a blank space as a separator to perform simultaneous searches. If you enclose a string of characters in quotation marks, the system searches for an exact match, including the blank spaces between the characters.

► To search for a member:

- 1 **Open a report and select a grid.**
- 2 **Double-click a dimension cell.**
- 3 **In the **Select Members** dialog box, next to **Find**, click the drop-down to select an option, and then enter the text for which you want to search.**
- 4 **Click .**


Tip: Use Wildcards is automatically enabled during member selection. When using wildcards, use ? to represent a single character search, and use * for a multiple character search.

If the search criteria has a space in it, enclose the search phrase in a double quote. For example, *IC Offset* should be "*IC Offset*". If the location of the "*" wildcard is at the end of the search string, for example IC Offset*, then you do not need to enclose the phrase in quotes.

Previewing Selected Members

You can preview the members or member lists selected for your report before you run it. The members that you view result from an evaluated member list or function.

► To preview members:


- 1 **Open a report and select a grid.**
- 2 **Double-click a dimension cell.**
- 3 **In the **Select Members** dialog box, in the **Selected** area, click .**

Copying Members Names From a Text Application

You can copy members names from a text application, such as an Excel spreadsheet, and paste them as new members in the corresponding dimension in Financial Reporting Web Studio.


For example, you could copy the members names in the “Year” dimension in an Excel spreadsheet to a grid in Financial Reporting Web Studio that contains the “Year” dimension.

► To copy members names from an application to Financial Reporting Web Studio:

- 1 **Open the source application and select the members to copy.**
- 2 **In Financial Reporting Web Studio, open a report and select a grid.**
- 3 **Select the dimension cell to which you want to copy members.**
- 4 **In the formula bar, clear the content, then right-click, and then select **Paste**.**
- 5 **Edit the text in the formula bar by adding commas between members.**
- 6 **Click  to verify the member names.**

Removing Members

You can remove members from a dimension. Removing members changes the output of a report, giving you more control over its content.


- To remove members that are contained in a single cell:
 - 1 Select the cell containing the members.
 - 2 Do one of the following:
 - Double-click the dimension name to display the **Select Members** dialog box, and then click  to remove the member from the Selected area of the dialog box.
 - Select the dimension in the grid, and then remove the member from the formula bar.

- To remove members that are in separate rows or columns:
 - 1 Right-click the row or column.
 - 2 Select **Delete**.

Selecting Multiple Members

You can select members dynamically based on criteria that you specify. You define criteria by creating expressions of members, boolean operations, and commands.

Boolean operators enable you to specify precise member combinations for the report, which is useful for dealing with large volumes of data. Use the **AND**, **OR**, **UNION**, and **NOT** boolean operators, combined with expression commands, to refine your member selections.

- To select multiple members based on criteria:
 - 1 Open a report and select a grid.
 - 2 Double-click a dimension cell.
 - 3 In the **Select Members** dialog box, in the **Members** tab, select at least two members, and then click .

You must select at least two members before you can create criteria.

- 4 In the Selected area of the dialog box, build expressions by using one or more of the following operators and symbols:
 - Select **Not** to add the Boolean operator **NOT** to the expression. **NOT** is the inverse of the selected condition.
 - Enter a left parenthesis, (, to add an opening character to the expression.
 - Enter a right parenthesis,), to add a closing character to the expression.
 - In the **Operator** column, select **And**, **Or**, or **Union**.

Tip: If you are using three members, use two pairs of parentheses. For example, if you select descendants of Market and you want to exclude East, West, and South, your advanced member selection query should be as follows:

```
Descendants of Market AND NOT (East AND NOT (West AND NOT SOUTH))
```

Access Privileges for Members in the Planning Database Source

Financial Reporting Web Studio gives you access privileges on members in the following dimensions:

- Accounts
- Entities
- Scenarios
- Versions

In the **Select Member** dialog box, all members for the Planning database connection are listed regardless of access privileges. Therefore, you can place members on a grid that are not returned when the report is run. The members you choose affect the output of the report. For example, if you select a member to which the user cannot access in the Point of View or Page Axis, an error message is returned.

The following four scenarios describe what values are returned on a grid that contains valid and invalid dimensions or members. These scenarios assume that you are reporting against a Planning database connection and have access privileges to the following dimensions and members:

- Accounts: Sales, Profit
- Versions: 1st Draft, 3rd Draft, Final Version
- Entities: North, Canada
- Scenarios: Budget

Scenario 1

You run a report based on the following grid:

Page: Budget

	1st Draft	2nd Draft	3rd Draft
North	###	###	###
Central	###	###	###
South	###	###	###

Because you do not have access to Central, South, and 2nd Draft, the report returns the following results:

Page: Budget

	1st Draft	3rd Draft
North	23.89	12.90

Scenario 2

You run a report based on the following grid:

Page: Actual

	1st Draft	2nd Draft	3rd Draft
North	###	###	###
Central	###	###	###
South	###	###	###

Because you do not have access to the Scenario dimension in the Pages axis of the grid, the member Actual is in the Scenario dimension, no page is displayed and an error message describing your access privileges is returned.

Scenario 3

You run a report based on the following grid:

Page: Budget

	1st Draft	2nd Draft	3rd Draft
London	###	###	###
Paris	###	###	###
New York	###	###	###

Because you do not have access to the entities on the rows, no page is displayed and an error message describing your access privileges is returned.

Scenario 4

You run a report based on the following grid:

Page: Budget

	Descendants of Versions
Descendents of Entities	###

The report returns the following data:

Page: Budget

	1st Draft	3rd Draft	Final Version
North	23.89	12.90	67.12
Canada	2.67	8.90	54.78

7

Formatting Reports

In This Chapter

Formatting Best Practices	75
Changing the Page Setup	75

Formatting reports enables you to specify how report components, such as text boxes, grids, images, and charts, appear in your reports. For example, you can specify fonts and add borders. When you format a report, you define the properties of the components. The formatting you apply affects the printed and online report presentation.

Note: Each object has default formatting properties. You can use the default format properties if you do not want to format your report.

Formatting Best Practices

When formatting a report, keep in mind the following points:

- Cell formatting has precedence over row and column formatting and grid formatting.
- Cell formatting allows you to make exceptions to row and column formatting and grid formatting.
- Row/column formatting has precedence over grid formatting.
- Row/column formatting allows you to make exceptions to grid formatting.
- The correct sequence in formatting a report is:
 1. Format the grid.
 2. Apply different formatting to rows and columns.
 3. Apply specific cell formatting.

Changing the Page Setup

You can alter page settings that control how reports are printed or viewed. You can change the paper size, page orientation, heading display settings, default width and length for the report, and create a custom size for the work area. You can also specify whether cell documents print with consecutive page numbers or print the page settings on each document.

► To change the page setup:

- 1 Open a report and select **File**, and then **Page Setup**.
- 2 Define information in the tabs in the **Page Setup** dialog box, and then click **OK**.

Note: When you print a report with cell document attachments enabled (see [“Attaching Cell Documents in a Report” on page 78](#)), the file attachment page numbers can be renumbered to reflect the page numbers of the report. For example, a nine page report with a five page Word document attached can renumber the Word document output to reflect the page number 10 to 15.

To print consecutive page numbers on cell documents, in the **Page Setup** dialog box on the **Page** tab, select **Consecutive Page Numbers**. To print the page numbers as specified on each cell document, clear **Consecutive Page Numbers**.



Providing For Detailed Data and Documents in Reports

In This Chapter

Setting Up Expansions to Access Detailed Data in Reports	77
Positioning Expansion Rows and Columns	77
Attaching Cell Documents in a Report	78

Setting Up Expansions to Access Detailed Data in Reports

You can design a report to display detail rows and columns of a grid at the request of a user. You do this by enabling the expansion of a dimension that has parent-child relationships among its members. For example, if you enable the Time dimension to expand, users can expand a row that displays quarters into months.

You access detail rows and columns by selecting summary rows and columns from reports that are displayed.

- To set up expansions to access detailed data in a report:
 - 1 Open a report and select a grid.
 - 2 Select the row or column headings for which you want to enable expansion.
 - 3 In **Heading Row Properties** or **Heading Column Properties**, select **Allow Expansion**.

Positioning Expansion Rows and Columns

You can specify whether expanded items are displayed before or after the dimension's summary-level row or column.

- To position expansion rows and columns:
 - 1 Open a report and select a grid.
 - 2 In **Grid Properties**, expand **Position**.
 - 3 In **Position of Expansions**, select an option:
 - **Before Parent**—Displays expansions before the summary-level row or column.

- **After Parent**—Displays expansions after the summary-level row or column. (This is the default.)

Attaching Cell Documents in a Report

Cell documents that are attached at the data source can be accessed in reports. The attached documents appear as hyperlinks at the end of a PDF or snapshot report. The documents are accessed in the HTML client when drilling into a cell. In addition, you can insert footnotes into a text object or text cell of the report that lists information about the attached cell documents by selecting the [ListofCellDocuments](#) text function. All cell documents have a point of view associated with the data cell to which they are attached.

A data cell can contain multiple cell documents. If a data cell contains cell documents in the data source, a designer can extract those documents, based on the given cell's POV. Cell documents can be attached to reports contained in books and snapshot books.

➤ To attach cell documents:

- 1 Select a *data type* cell in a grid.
- 2 In **Cell Properties**, select **Retrieve Cell Documents**.

For optimum retrieval performance, select **Retrieve Cell Documents** only for cells with attachments to include in the report. If cell documents exist for the selected cell, they are appended to the report when printed.

When you attach cell documents in a report, consider the following points:

- Include the correct supported extensions for files.
- Files which are corrupt or improperly uploaded to the ADM provider will not print.
- Documents containing file extensions that are not properly registered to print will fail.
- Oracle Hyperion Financial Reporting Web Studio does not support Microsoft files containing macros or that trigger UI components requiring user input. Some Microsoft files with macros perform operations when the file is opened which include the launching of dialogs requiring user input. Files that display dialogs or UI components requiring user input may not be available to print.

This may also hold up the completion of the PDF generation process. The server's administrator may need to stop the application through Task Manager to resume printing activities. Because the application is launched under the SYSTEM account, it may not be visible since the process was not initiated by the login account.

Supported Microsoft Office File Types

- Microsoft Word files (*.doc, *.docx, *.rtf)
- Microsoft Excel files (*.xls, *.xlsx)
- Microsoft PowerPoint files (*.ppt, *.pptx)

- PDF files (*.pdf)
- Text files (*.txt)

Viewing Reports Containing Cell Documents

In Planning and Budgeting Cloud Workspace, you can view reports containing cell documents in PDF or HTML output.

PDF Output

- Hyperlinks to all files are included in the bottom of the document output, after the Financial Reporting report content. Click a hyperlink to launch the document.
- The function returns the file name and description parameters for all cell documents. The page number parameter is only displayed for the supported Microsoft Office documents.

HTML Output

- All cell documents in HTML preview are hyperlinks.
- Depending on the user settings in the browser, the cell document may be opened automatically in a new browser window.
- The [ListofCellDocuments](#) function returns the file name and description parameters, but not the page number parameter.

Displaying a List of Cell Document

Use the [ListofCellDocuments](#) text function to display a list of all cell documents and attributes such as description and start page number that are retrieved in the report. You can enter the function in a text object, text cell, or text cell heading.

9

Previewing and Printing Reports

In This Chapter

Previewing Reports	81
Responding to Prompts When Previewing	81
Printing Tips for Designers.....	82

Previewing Reports

➤ To preview a report in a web browser:

- 1 Open a report in the Report Designer.
- 2 Do one of the following:

- Click  or .
- Select **File**, and then either **HTML Preview** or **PDF Preview**.

Responding to Prompts When Previewing

If a report is designed with prompts, those prompts are displayed when you preview the report. You must respond to the prompt and provide the requested information by selecting members from the prompt list. You can also edit the prompts manually. If the prompts contain alias names, you can edit the alias names by converting to member names. See [“Setting Up Report Prompts” on page 67](#).



➤ To respond to a prompt when previewing a report:

- 1 With a report open in the Report Designer, select **File**, and then either **HTML Preview** or **PDF Preview**.
- 2 In the **Respond to Prompts** dialog box, under the **Selection** column, do one of the following:
 - If the member is known, enter the member name in the text box for the respective prompt.

If the text box is disabled, the prompt contains alias names. To edit the text box, select **Edit Member Names**. If multiple members are provided for the prompt, separate the members with commas.

Selecting **Edit Member Names** displays member names in the text box, not alias names. You can edit the member name associated with an alias.

- Click .

In the **Select Members** dialog box, the default member is listed in the Selected area. Click  and  to move members between the panels.

For more information on defining members, see [Chapter 6, “Defining Members,”](#)

- 3 Select **OK**, and then click **Run**.

Printing Tips for Designers

Subtopics

- [Positioning and Alignment](#)
- [Printing Text Boxes](#)
- [Printing Grids](#)
- [Printing Page Dimension Headings](#)
- [Printing Headers and Footers](#)

Positioning and Alignment

Setting relative positioning on auto-sized report objects gives you precise control over how report objects are positioned relative to one another. If none of the report objects in your report is auto-sized, then the relative positioning is ignored.

If you set the position of a report object to relative, all report objects within the design shift so that the distance between the auto-sized report object and fixed report objects remain the same.

For example, if Grid A is auto-sized and has horizontal positioning set to “center”, then Grid A is printed in the center of the page. If Chart A is linked to Grid A and has its vertical position set to “relative,” then Chart A is printed one inch from the bottom edge of Grid A.

If the alignment for Chart A is not set to relative, it is printed on the absolute coordinates in the layout of the grid, so it might overwrite Grid A.

Note: Vertical and horizontal positioning are not maintained after changes are made to the report.

Note: When printing reports containing overlapping objects, objects may not print as displayed in the designer, requiring you to rearrange the objects in your report.

Printing Text Boxes

If you set the auto-size property for a text box:

- when you print a report, the height of a text box increases or decreases to accommodate all the text in the text box.
- The width of the text box does not change.

If the auto-size property is not set when you print a report:

- The text box is printed at the height and width of the grid.
- Text that does not fit in the text box is truncated.

Printing Grids

If you set the auto-size property for a grid and then print the grid, the height and width of the grid increases or decreases to accommodate all data in the grid.

When the auto-size property is not set, the grid prints at the height and width specified. If a row or column does not fit on a page, it prints on subsequent pages. The columns print first using as many pages as needed, followed by the rows.

You use the positioning properties of the grid to align the grid on the page. Manually positioning the grid to the right or bottom of the page does not ensure that the grid will print on the right or bottom of the report.

When a grid is set to auto-size, and does not fit on one page, it prints on subsequent pages. The columns print first, starting on the left margin of the next page, followed by the rows, which print starting at the top of the next page.

Printing Page Dimension Headings

You can specify a page dimension heading for a grid in **Page Properties**. When you print a report, the page dimension heading prints every time the page member changes.

Page dimension headings print as follows:

- The text is left justified at the left edge of the grid.
- The text is auto-sized vertically, and the width is set to the width of the grid.
- The heading prints above the first row in the grid each time the page member changes.

A page dimension heading prints the member name every time the page member changes. Members on the page dimension can have their own custom heading.

➤ To define a page dimension heading:

- 1 Open a report and select a grid.
- 2 Select the **Pages** list box on the grid.
- 3 In **Page Properties**, under **Page Printing Positioning**, select an option for the heading.
 - **New Page - Top**—Prints a new page at the top of the next page.

Use this option when a chart is linked to a grid and you want to ensure that the chart and grid print next to each other every time the page member changes.

- **New Page - Same**—Prints a new page at the specified grid location.

Use this option in the following situations:

- The grid has horizontal or vertical alignment properties set, and you want the grid to print in the same position on each page. For example, if you set the grid position to Center/Middle, it prints in the center of each page.
 - When a chart is linked to a grid, and you want to ensure that the chart and grid print next to each other whenever the page member changes.
- **Same Page**—Prints the new page immediately after the data from the previous page.

Note: If a chart is linked to a grid, the chart prints once; not every time the page member changes.

Printing Headers and Footers

When printing header and footers, keep in mind the following points:

- You can place an auto-sized text box in the header; however, if the text box height expands beyond the header, it prints beyond the header boundary.
- You can place an auto-sized text box in the footer; however, if the text box height expands beyond the footer, the text is truncated.
- If you place an object near the top or bottom of the page, it might overlap the header or the footer.

10

Using Functions

In This Chapter

Mathematical Functions	86
Text Functions	103
Conditional Functions	132
Financial Functions	135

Mathematical Functions

Subtopics

- [About Mathematical Functions](#)
- [Abs](#)
- [Average](#)
- [AverageA](#)
- [CountA](#)
- [Difference](#)
- [Eval](#)
- [Max](#)
- [Min](#)
- [PercentofTotal](#)
- [Product](#)
- [Round](#)
- [Sum](#)
- [Truncate/Truc](#)

About Mathematical Functions

Subtopics

- [Arguments in Mathematical Functions](#)
- [Mathematical Operators in Expressions](#)
- [Mathematical Operands in Expressions](#)
- [Natural Precedence](#)

Mathematical functions perform calculations on numeric values or data from a grid. Mathematical functions, formulas, and their syntax are not case-sensitive.

The syntax for a mathematical function is:

FunctionName(*arguments*)

where:

FunctionName is the name of the mathematical function.

arguments are a numeric value, a row, column, or cell reference, or an embedded function.

Arguments in Mathematical Functions

Subtopics

- [Numeric Arguments](#)
- [Row, Column, or Cell Reference Arguments](#)
- [Aggregate Property Arguments](#)
- [Reference Property Arguments](#)
- [Embedded Functions as Arguments](#)

Numeric Arguments

The syntax for a numeric argument is:

(numeral1, numeral2, ... numeraln)

where numerals 1 through *n* are any numbers including decimals and negative values. For example, the expression `Average(10,20,30)` returns the value 20.

Row, Column, or Cell Reference Arguments

The syntax for a row, column, or cell argument is:

FunctionName(GridName.GridElement[segment(range)].Property)

Table 6 Argument Components

Argument	Description
<i>GridName</i>	(Optional) Name of a grid. For example, <code>Difference(grid1.row[5], grid2.row[5])</code> returns the difference of two rows on grid1 and grid2. If <i>GridName</i> is not specified, the default is the current grid where the formula is entered.
<i>GridElement</i>	(Optional) One of the following keywords: <code>row</code> , <code>col</code> , <code>column</code> , or <code>cell</code> . For example, <code>Max(row[1], row[2], row[3])</code> returns the maximum value of three rows. (The keyword <code>row</code> and <code>column</code> or <code>col</code> is optional.) You must specify row and column segment IDs. For example, <code>cell[2, A]</code> refers to the cell that is the intersection between row 2 and column A. The keyword <code>cell</code> is optional. For example, <code>[5, B]</code> refers to the cell that is the intersection between row 5 and column B. Cell references can use <code>[row, col]</code> syntax or <code>[col, row]</code> syntax. If <i>GridElement</i> is specified, letters represent columns and numbers represent rows; for example, <code>Max([1,A], [2,A], [3,A])</code>
<i>segment</i>	(Required) Row, column, or cell reference number of a grid. For an expanded row or column, you must specify the segment. For example, <code>row[2]</code> addresses row segment 2. Segments are enclosed in square brackets [].

Argument	Description
<i>range</i>	<p>(Optional) Rows, columns, or cells that are expanded from the specified segment. If <i>range</i> is specified, the formula is calculated using only the specified range. For example, <code>row[2 (3:5)]</code> uses only the 3rd through 5th rows of expanded segment 2.</p> <p>When <i>range</i> is not provided, all expanded cells are used.</p> <p>Note: If a segment expands to only one row or column, do not use the <i>range</i> argument.</p>
<i>property</i>	<p>(Optional) One of the following keywords: <code>average</code>, <code>averageA</code>, <code>count</code>, <code>countA</code>, <code>max</code>, <code>min</code>, <code>product</code>, or <code>sum</code>. Property is used to aggregate the expanded rows, columns, or cells specified.</p> <p>Do not specify property when a reference is an argument. By not specifying property, the function calculates the reference in the most appropriate way. For example, the following expression returns the average of the cells within rows 1 and 2:</p> <pre>Average(row[1], row[2])</pre> <p>Whereas the following example first calculates the average of <code>row[1]</code>, then the average of <code>row[2]</code>, adds these two results, then divides by 2:</p> <pre>Average(row[1].average, row[2].average)</pre> <p>The default property for a row, column, or cell reference that is <i>not</i> used as a function argument is the <code>sum</code> property. For example, the following expression's default property is <code>sum</code>:</p> <pre>row[2]</pre> <p>See “Aggregate Property Arguments” on page 88 and “Reference Property Arguments” on page 89.</p>

Because *segment* is the only required part of a reference, the following references are the same:

```
Grid1.row[1].sum  
[1]
```

`AverageA` and `CountA` include `#missing` and `#error` cells in the calculation. For example, if row 1 is a segment row that expands to `Qtr1 = 100`, `Qtr2 = 200`, `Qtr3 = #missing`, and `Qtr4 = 400`, the following function returns the value four (4):

```
row[1].CountA
```

All other functions exclude `#missing` data or `#error` cells. For example, the previous example of row 1 that expands to `Qtr1 = 100`, `Qtr2 = 200`, `Qtr3 = #missing`, and `Qtr4 = 400`, returns three in this example:

```
row[1].Count
```

Aggregate Property Arguments

An aggregate row, column, or cell contains multiple rows, columns, or cells, respectively. The aggregate property argument is the last argument in the following mathematical function syntax:

```
FunctionName(GridName.GridElement[segment(range)].property)
```

You apply aggregate properties to a row, column, or cell reference. Aggregate properties include:

- [Average](#)
- [AverageA](#)
- [CountA](#)

- [Max](#)
- [Min](#)
- [Product](#)
- [Sum](#)

When used as a mathematical function argument, the default for property is the same as the function. In the following example the default property is *Average*:

```
Average(row[2])
```

When not used as a mathematical function argument, the default for property is *sum*. In the following example the default property is the sum of an aggregate row:

```
row[2]
```

Reference Property Arguments

A reference property argument specifies how to treat formula reference results and is used in conjunction with the other properties.

There is one reference property argument: *IfNonNumber* / *IFFN*.

IfNonNumber specifies a replacement for *#Missing* and *#Error* values with a specific numeric value.

The syntax is:

```
AXIS[segment(range)].IfNonNumber(arg).AggregateProperty
```

Table 7 Argument Components

Argument	Description
AXIS	(Optional) A row, column, or cell keyword.
<i>Segment(range)</i>	Any axis reference, such as a row number or column letter.
<i>IfNonNumber</i>	How to treat missing or error data within the <i>AxisRef</i> .
<i>(arg)</i>	What number to use if missing or error data is encountered within the <i>AxisRef</i> .
<i>AggregateProperty</i>	(Optional) The aggregate function used for aggregate segments. See “Aggregate Property Arguments” on page 88 .

For example:

If *cell[1,A] = 3* and *cell[1,B] = #Missing*,

The following expression returns *#Error*:

```
cell[1,A] / cell[1,B]
```

The following expression replaces *cell[1,B]* with 1 and returns 3:

```
cell[1,A] / cell[1,B].ifNonnumber(1)
```

Note: If you use suppression for #Missing or #Error in a grid, and the grid contains a formula row or column that uses the IfNonNumber property, #Missing and #Error remain suppressed.

Embedded Functions as Arguments

You can embed functions as arguments within a function. In the following example, the function Average is embedded in the function Sum:

```
sum(row[3:5], avg(row[4:6], 40, 50), row[7; 9], 70, 80)
```

- Row segments 3, 4 and 5
- The average of row segments 4, 5 and 6, with the numbers 40 and 50
- Row segments 7 and 9
- The numbers 70 and 80

Mathematical Operators in Expressions

Operators are symbols that perform arithmetical tasks or comparisons, or refer to ranges of columns, rows, or cells. Use operators in formulas that perform calculations on data rows or columns.

Table 8 Operators in Standard Formulas

Type	Operator	Description	Example
Arithmetical	+	Add	Add the values in rows 4 and 5. [4] + [5] A space is needed after + . <ul style="list-style-type: none"> ● Incorrect: 5+4 ● Correct: 5+ 4 ● Correct: 5+ -4
Arithmetical	-	Subtract	Subtract 3 from the values in row 4. [4] - 3. A space is needed after - . <ul style="list-style-type: none"> ● Incorrect: 5-4 ● Correct: 5 - 4 ● Correct: 5 - -4
Arithmetical	*	Multiply	Multiply the absolute values in row 4 by 150. Abs([4]) * 150
Arithmetical	/	Divide	Divide the values in row 4 by the values in row 5. [4] / [5]

Type	Operator	Description	Example
Arithmetical	%	Percentage	Add the values in row 4 and multiply it by 10. <code>row[4].Sum*10%</code>
Arithmetical	^	Exponential	Return the sixth exponent of row 4. <code>[4]^6</code>
Arithmetical	Decimal	Decimal number	Multiply the values in row 4 by 5.67. <code>[4] * 5.67</code>
Reference	[]	Specifies a row, column, or cell in a grid. Use numbers for rows and letters for columns.	Add the values in rows 4 and 12. <code>Sum ([4], [12])</code>
Unary minus	-	Changes the sign of a value	Divide the values in column 4 by 12 and change the sign of the result. <code>([4] / 12)</code>

Tip: Use the `Eval` function to perform arithmetic operations as part of a function parameter. See [“Eval” on page 96](#).

Tip: Natural precedence determines the order to carry out operations in expressions with multiple operators. See [“Natural Precedence” on page 92](#).

Mathematical Operands in Expressions

Operands specify the values that an operator uses to produce a result.

Table 9 Operands Supported in Financial Reporting Web Studio

Operand	Examples
Literal	3, 0.0, 27.5, 65.334, -841
Row or column reference number	[1], [4], [8], [A], [C:D]
Cell reference number	[2, E], [E, 2] Note: This cell reference addresses row 2 column E.
Functions	Average, Min, Max
Scoping	Grid1.row[3]

In the following row formula, which multiplies the values in row 3 by 100, the row reference [3] and 100 are both operands:

`[3] * 100`

Natural Precedence

If a standard formula has two or more operators, the system performs the operations in the order of the operators' natural precedence.

The order of natural precedence to perform operations in an expression that has multiple operators is:

1. Unary minus (-# where # is any number)

Note: Do not add a space after the unary minus operator.

2. Multiplication (*) and division (/)
3. Subtraction (- #) and addition (+ # where # is any number)

Note: Add a space after the subtraction and addition operator.

For example, if row 10 has a value of 8, the following expression produces a value of 20 for that column by calculating $3 * 4 + 8 = 20$, following the order of natural precedence:

[10] + 3 * 4

Abs

Abs is a mathematical function that returns the absolute value of a numeric value, row, column, or cell. The absolute value of a number is that number without a negative sign. A negative number becomes positive, while a positive number remains positive.

Syntax:

Abs (*argument*)

where *argument* is one of the following values:

Argument	Description
numeric	A numeric value. For example, Abs(-20) returns the value 20. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <i>GridName</i> . <i>GridElement[segment(range)].Property</i> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Examples:

Return the value 30:

Abs(-30)

Return the absolute value of the value in row 1:

```
Abs (row[1])
```

Calculate the absolute value of the sum of column E:

```
Abs (column[E].sum)
```

Point to expanded rows 1 through 3 within design segment 3 of Grid1:

```
Abs (Grid1.row[3(1:3)])
```

Average

Average is a mathematical function that returns the average of a group of numeric values, rows, columns, or cells. Average excludes #missing and #error cells when obtaining the average.

Note: The calculation does not include missing values regardless of whether they are suppressed.

Syntax:

```
Average (arguments)
```

or

```
Avg (arguments)
```

where *arguments* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, <code>Average(10, 20, 30)</code> returns the value 20. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. For example <code>Avg(Grid1.row[4(3:5)])</code> returns the average of grid1, row segment 4, range 3 through 5. You can specify references in several ways. The reference syntax is: <code>GridName.GridElement[segment(range)].Property</code> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Examples:

Return the value 20:

```
Avg(10, 30, 20)
```

Return the average of all numbers that are part of three aggregate rows:

```
Average(row[1], row[6], row[8])
```

Calculate the average of three aggregate columns; E, G, and I. (The calculation produces three numbers, then calculates the average of the three numbers.)

`Avg (column [E] avg, column [G] avg, column [I]. avg)`

Calculate the average of two columns; E on the current grid and E on grid2. (The calculation includes all expanded columns.)

`Avg (column [E], grid2column [E])`

Calculate the average of aggregate row 3 and divide the average by 100:

`Avg (row [3]) / 100`

AverageA

AverageA is a mathematical function that returns the average of a group of numeric values, rows, columns, or cells. AverageA includes #missing and #error cells, which are treated as zero values when obtaining the average.

Note: #missing and #error are included only for rows or columns that are not suppressed.

Syntax:

`AverageA (arguments)`

or

`AvgA (arguments)`

where *arguments* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, <code>AverageA (10, 20, 30)</code> returns the value 20. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <code>GridName.GridElement [segment (range)].Property</code> . See “Row, Column, or Cell Reference Arguments” on page 87 . For example, <code>AvgA (Grid1.row [4 (3:5)])</code> returns the average of grid1, row segment 4, range 3 through 5.
function	An embedded function. See “Mathematical Functions” on page 86 .

Example:

If a grid has four rows with the values 10, 20, 30, and #error, the following formula in the 5th row returns the value 15:

`AverageA ([1:4])`

CountA

CountA is a mathematical function that returns the number of values in a group of numeric values, rows, columns, or cells. CountA includes #missing and #error cells when obtaining the count only for rows or columns that are not suppressed.

Syntax:

CountA(*arguments*)

where *arguments* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, CountA(10,20,30,50) returns the value 4. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: CountA(GridName.GridElement[segment(range)].property). See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Examples:

If a grid has four rows with the values 10, 20, 30, and #error, the following formula in the 5th row returns the count of four rows:

```
CountA([1:4])
```

Return the count of four rows:

```
CountA(row[1], row[6], row[8] row[where data yields #error])
```

Difference

Difference is a mathematical function that returns the absolute value of the difference of a numeric value, row, or column subtracted from another numeric value, row, or column.

Syntax:

Difference(*arg1*, *arg2*)

where *arg2* is subtracted from *arg1* and is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, Difference(3,5) returns the absolute value 2. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .

Argument	Description
row, column, or reference	<p>A pointer to a row, column, or cell within a grid.</p> <p>You can specify references in several ways. The reference syntax is: <i>GridName</i>. <i>GridElement[segment(range)].Property</i>. See “Row, Column, or Cell Reference Arguments” on page 87.</p> <p>The following example returns the difference of two rows on grid1 and grid2:</p> <pre>Difference(grid1.row[1], grid2.row[6])</pre>
function	<p>An embedded function.</p> <p>See “Mathematical Functions” on page 86.</p>

Note: `Difference` returns the absolute value of *arg2* subtracted from *arg1*, whereas the minus sign in subtraction negates a number.

Examples:

Return the absolute value of 8:

```
Difference(3, -5)
```

Calculate the difference of two aggregate columns:

```
Difference(column[E], column[G])
```

Calculate the difference of two columns that are located on different grids, grid1 and grid2:

```
Difference(grid1.column[E], grid2.column[E])
```

Note: You can type the text label “Difference” or “Variance”.

Eval

`Eval` is a mathematical function that evaluates an expression. Use `Eval` as an embedded function argument to consolidate multiple expressions into one expression.

Syntax:

```
Eval(expression)
```

where *expression* is one or more of the following values:

Argument	Description
numeric	<p>A numeric value.</p> <p>Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87.</p>
row, column, or reference	<p>A pointer to a row, column, or cell within a grid.</p> <p>References can be specified in several ways. The reference syntax is: <i>GridName</i>. <i>GridElement[segment(range)].Property</i>. See “Row, Column, or Cell Reference Arguments” on page 87.</p>

Argument	Description
function	An embedded function. See “Mathematical Functions” on page 86 .
operators	Any of the supported arithmetic operators (+, -, *, /, ^, %).

Example:

Divide row 1 by row 2, and then round the data to four places:

```
Round (Eval ([1] / [2]), 4)
```

Note: In this example, you must use `EVAL` to get the desired result. The following does not work:

```
Round (([1] / [2]), 4).
```

Max

`Max` is a mathematical function that returns the maximum value in a group of numeric values, rows, columns, or cells.

Syntax:

```
Max (arguments)
```

where *arguments* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, <code>Max(10, 20, 30)</code> returns the value 30. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <code>GridName.GridElement[segment(range)].Property</code> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Examples:

Return the maximum value in rows 1, 6, and 8:

```
Max(row[1], row[6], row[8])
```

Calculate the maximum of the sums of aggregate rows:

```
Max(row[1].sum, row[2].sum, row[3].sum)
```

Min

Min is a mathematical function that returns the minimum value in a group of numeric values, rows, columns, or cells.

Syntax:

Min(*arguments*)

where *arguments* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, Min(10,20,30) returns the value 10. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <i>GridName</i> . <i>GridElement[segment(range)].Property</i> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Examples:

Return the minimum value in rows 1, 6, and 8:

```
Min (row[1], row[6], row[8])
```

Calculate the minimum of the sums of aggregate rows:

```
Min(row[1].sum, row[2].sum, row[3].sum)
```

PercentofTotal

PercentOfTotal is a mathematical function that returns the result of a numeric value, row, column, or cell divided by another numeric value, row, column, or cell which is multiplied by 100.

Syntax:

PercentOfTotal (*arg1*, *arg2*)

where:

- *arg1* is a component of the running total (*arg2*), usually a row or column reference.
- *arg2* is the running total relative to *arg1*, usually a cell reference containing the grand total.
- *arg1* is divided by *arg2* with the result multiplied by 100. *Arg1* and *arg2* are one or more of the following values:

Argument	Description
numeric	A numeric value. For example, <code>PercentofTotal(100,20)</code> returns the value 500. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <code>GridName.GridElement[segment(range)].Property</code> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Examples:

Return the value of 5 percent.

```
PercentofTotal(20,400)
```

Divide the value of each cell in column A by the total market value in cell A5, multiply the result by 100, and display the resulting `PercentOfTotal` in column B.

```
PercentOfTotal([A],[A,5])
```

Using the above example, the following table shows the `PercentOfTotal` results in column B:

		A	B
1		Sales	% Total
2	Mkt1	60	20%
3	Mkt2	120	40%
4	Mkt3	120	40%
5	Total Mkt	300	100%

Tip: You enter the formula by clicking on the header for Column B and using the formula bar.

Product

`Product` is a mathematical function that multiplies all numbers or references and returns the product.

Syntax:

```
Product(arguments)
```

where *arguments* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, <code>Product(2,20)</code> returns the value 40. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <code>GridName.GridElement[segment(range)].property</code> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Example:

Return 40:

`Product(2,20)`

Round

Round is a mathematical function that rounds a number up or down by the specified digits.

Syntax:

`Round (arg1, integer)`

where *arg1* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, <code>Round(81.3987,3)</code> returns the value 81.399. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <code>GridName.GridElement[segment(range)].property</code> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Integer specifies the number of digits to round the number:

- If *integer* is greater than zero, the number is rounded to the specified number of decimal places.
- If *integer* is zero, the number is rounded to the nearest integer.
- If *integer* is less than zero, the number is rounded to the left of the decimal point.

Examples:

Round to 3 decimals:

`Round(3594.5567, 3) = 3594.557`

Round to the nearest integer:

`Round(3594.5567, 0) = 3595`

Round to the thousand, (also known as scaling):

`Round(3594.5567, -3) = 4000`

Sum

Sum is a mathematical function that returns the summation of a group of numeric values, rows, columns, or cells.

Syntax:

`Sum(arguments)`

where *arguments* is one or more of the following values:

Argument	Description
numeric	A numeric value. For example, <code>Sum(10, 20, 30)</code> returns the value 60. Numeric values can include decimals and negative values. See “Numeric Arguments” on page 87 .
row, column, or cell referenc	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <code>GridName.GridElement[segment(range)].property</code> . See “Row, Column, or Cell Reference Arguments” on page 87 .
function	An embedded function. See “Mathematical Functions” on page 86 .

Examples:

Return the value 30:

`sum(10, 20)`

Return the sum of three rows:

`sum(row[1], row[6], row[8])`

Calculate the sum of three aggregate columns:

`sum(column[E], column[G], column[I])`

Calculate the sum of two columns that are located on different grids:

`sum(grid1.col[E], grid2.colmn[E])`

Truncate/Truc

Truncate is a mathematical function that removes the specified number of digits from numeric values.

Syntax:

`Trunc (arg1, integer)`

where *arg1* is one of the following values:

Argument	Description
numeric	A numeric value. For example, 234.567. See “Numeric Arguments” on page 87.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. You can specify references in several ways. The reference syntax is: <i>GridName</i> . <i>GridElement[segment(range)].property</i> . See “Row, Column, or Cell Reference Arguments” on page 87.
function	An embedded function. See “Mathematical Functions” on page 86.

Integer specifies the number of digits to remove:

- A positive *integer* determines the number of digits to the right of the decimal point.
- A zero (0) *integer* returns the integer located to the left of the decimal point.
- A negative *integer* indicates the number of digits to the left of the decimal point.

Examples:

The following statement uses a positive integer of 2. The first two digits to the right of the decimal point remain, and the following digit is removed:

`Trunc (234.567, 2) = 234.56`

The following statement uses a zero (0) integer. All digits to the right of the decimal point are removed:

`Trunc (234.567, 0) = 234`

The following statement uses a negative integer of -2. All digits to the right of the decimal point are removed and the last 2 digits of the integer are truncated.

`Trunc (234.567, -2) = 200`

Note: Any formatting previously applied to a cell, column, or row is maintained when you use the `Trunc` function. The following example shows the results of a `Trunc` function where the cell value was previously formatted to display three decimal places: `Trunc (234.567, 0) = 234.000`

Text Functions

Subtopics

- [About Text Functions](#)
- [Annotation](#)
- [CellText](#)
- [Date](#)
- [DataSource](#)
- [Footnote](#)
- [GetCell](#)
- [GetHeading](#)
- [GridDimension](#)
- [ListofCellDocuments](#)
- [MemberAlias](#)
- [MemberName](#)
- [MemberProperty](#)
- [Page](#)
- [PageCount](#)
- [PlanningAnnotations](#)
- [ReportAuthor](#)
- [ReportCreated](#)
- [ReportDesc](#)
- [ReportFolder](#)
- [ReportModified](#)
- [ReportModifiedBy](#)
- [ReportName](#)
- [ReportRunBy](#)
- [RetrieveValue](#)

About Text Functions

Subtopics

- [Text Function Considerations](#)
- [Using the Current/Cur Keyword in a Text Function](#)

Text functions return report information, such as the report name or the report description. You use text functions in grids or in text objects.

Text Function Considerations

- Enclose text functions in braces:

```
<<TextFunction (arguments)>>
```
- You can use spaces between arguments.

- All arguments are enclosed in quotes, except for numeric arguments.

For example, the parameter “NYC, New York” in the function
`<<MemberName ("Grid1", NYC, New York)>>` should be restated as
`<<MemberName ("Grid1", "NYC, New York")>>`.

- Arguments related to date formatting are case-sensitive.
- To apply text functions to a grid, create a text row or column, or a heading cell, and then insert the text function.

Using the Current/Cur Keyword in a Text Function

You can use the `current` keyword with certain text functions to make the formulas more dynamic, more useful, and less redundant. The `current` keyword can replace required parameters in text functions. Hard-coded parameters produce the same results, whereas the `current` keyword can produce dynamic values. For example, when you use `current` for the `Page` parameter in a multiple page grid, it produces the value for each page.

You can use any form of the `current` keyword; `current`, or `cur`. Current keywords are not case-sensitive.

Note: The `current` keyword is supported in a text object only when using the following text functions: `MemberName`, `MemberAlias`, and `MemberDescription`, where `current` specifies the `Page` dimension parameter. However, the `Grid` name, `Row`, and `Column` must be defined, where applicable, and cannot use the `current` keyword.

Example 1:

In the `GetCell` function, many of the parameters are or may be relative to the current cell in which the function is entered. Therefore, the `current` keyword can be applied to this function. The `GetCell` function uses the following syntax:

```
<<GetCell ("GridName", Row, Column, Page)>>
```

In the following example, the hard-coded parameters for `GetCell` return the same cell value of Grid 1, cell 1A, page 1 across multiple pages:

```
<<GetCell ("Grid1", 1, A, 1)>>
```

The next example uses the `current` keyword with the `GetCell` function. The cell where the function is entered displays different values for each page. Notice that the `current` keyword is also applied to the grid name. If the grid name is changed, the text formula is still correct.

```
<<GetCell ("current", 1, current, current)>>
```

Example 2

When you use the `current` keyword in a cell that expands, the results are relative to the expanded cell.

The following example uses `GetCell` functions in an expanded cell (Column A expands to columns Qtr1, Qtr2, Qtr3 and row 1 expands to rows East and West). Text Row 1 uses the

current keyword for the Column parameter; Text Row 2 uses a hard-coded value for the Column parameter.

Name	Description
	Qtr1, Qtr2, Qtr3
East, West	#
Text Row 1	<<GetCell(cur, 1, cur, cur)>>
Text Row 2	<<GetCell(current, 1, A, 1)>>

The results are displayed in the following table.

Name	Quarter 1	Quarter 2	Quarter 3
East	5,120	4,502	7,304
West	3,405	2,300	4,462
Text Row 1	5,120	4,502	7,304
Text Row 2	5,120	5,120	5,120

- Text Row 1 uses the current keyword for the Column parameter, which results in the update of the current column of each expanded cell (Qtr1, Qtr2, Qtr3). Also, because the current keyword is used for the Page parameter, the values in Text Row 1 update each page relative to the current page of the grid.
- Text Row 2 uses a hard-coded value for the Column parameter (A), which results in the same value of 5,120 for all quarters. A hard coded reference to a Column parameter returns the same top left value of the column and row.

Table 10 Text Functions that Support the Current Keyword

Text Function	Example
<pre><<DataSource("GridName.Axis[ID]", InfoType)>></pre> <p>Parameters where applicable:</p> <ul style="list-style-type: none"> ● GridName: supported in all cells ● InfoType: not applicable 	<pre><<DataSource(current, "App")>></pre> <p>Note: The current grid name is used.</p>

Text Function	Example
<p><<GetCell("GridName", Row, Col, Page)>></p> <p>Parameters where applicable:</p> <ul style="list-style-type: none"> ● GridName: supported in all cells ● Row: supported in all cells ● Col: supported in all cells ● Page: supported in all cells 	<p><<GetCell(cur, 1, cur, current)>></p> <p><<GetCell(cur, 1(3), cur, current)>></p> <p>Note: The value from the third (expanded) row of row 1 is shown.</p> <p><<GetCell(cur, cur, A, current)>></p> <p><<GetCell(cur, cur, A(B), current)>></p> <p>Note: The value from the second (expanded) column of column A would be shown.</p>
<p>Use the following in a data row or column heading:</p> <p><<MemberName("DimName")>></p> <p>Use the following in a non-data row or column heading:</p> <p><<MemberName("GridName", Row/Col/Page, "DimName")>></p> <p>Use the following parameter where applicable:</p> <p>DimName: Heading cells only</p>	<p><<MemberName(current)>></p> <p><<MemberName("current", Row/Col/Page, "current")>></p> <p>Note: Returns the name of the member of the data row, column, or page that corresponds to the dimension where the heading cell is located. This can be used with the <<MemberDimension>> function to create custom headings for a grid.</p>
<p><<CalcStatus("GridName", Row, Col, Page)>></p> <p>Note: The parameters allowing the <i>current</i> keyword are highlighted in bold.</p>	<p><<CellText("GridName", Row, Column, Page)>></p> <p><<CellText("GridName", Row, Column, Page, MemberOverride)>></p>

Text Function	Example
<pre><<GridDimension("GridName", "Axis", index)>></pre> <p>Note: The parameters allowing the <code>current</code> keyword are highlighted in bold.</p>	<pre><<MemberAlias("DimName")>> <<MemberAlias("GridName", "DimName")>> <<MemberAlias("GridName", Row Col Page, "DimName")>> <<MemberDesc("DimName")>> <<MemberDesc("GridName", "DimName")>> <<MemberDesc("GridName", Row Col Page, "DimName")>> <<MemberName("DimName")>> <<MemberName("GridName", "DimName")>> <<MemberName("GridName", Row Col Page, "DimName")>> <<MemberQualifiedName("DimName")>> <<MemberQualifiedName("GridName", "DimName")>> <<MemberQualifiedName("GridName", Row Col Page, "DimName")>> <<MemberProperty(DimensionName, Property)>> <<MemberProperty("GridName", DimensionName, Property)>> <<MemberProperty("GridName", Row Col Page, DimensionName, Property)>></pre>
<pre><<ProcessManagementStatus("GridName", Row, Column, Page)>></pre> <p>Note: The parameters allowing the <code>current</code> keyword are highlighted in bold.</p>	
<pre><<ListOfCellDocuments("GridName", FileName)>></pre> <p>Note: The parameters allowing the <code>current</code> keyword are highlighted in bold.</p>	
<pre><<GridDimension("GridName", "Axis", Index)>></pre> <p>Parameters, where applicable:</p> <ul style="list-style-type: none"> ● GridName: supported in all cells ● Axis: Heading cells only ● Index: Heading cells only 	<pre><<GridDimension(cur, "Page", 1)>> <<GridDimension(cur, cur, cur)>></pre> <p>Note: The dimension name corresponding to the heading cell is shown.</p>

Text Function	Example
<pre data-bbox="159 218 792 302"><<GetHeading("GridName", Page, Ref, Offset)>></pre> <pre data-bbox="159 273 792 302"><<GetHeading("GridName", Page, Ref, "Delim")>></pre> <p data-bbox="159 327 435 357">Parameters where applicable:</p> <ul data-bbox="159 373 860 567" style="list-style-type: none"> ● <i>GridName</i>: supported in all cells ● <i>Page</i>: supported in all cells ● <i>Ref</i>: supported for in all cells, but should not be used in heading cells* ● <i>Offset</i>: Heading cells only ● <i>Delim</i>: not applicable 	<pre data-bbox="993 218 1490 247"><<GetHeading(cur, cur, A, " - ")>></pre> <pre data-bbox="993 273 1490 302"><<GetHeading(cur, cur, 1, " - ")>></pre> <p data-bbox="993 327 1490 386">Note: In the above examples, all headings in column A and row 1 are shown for the current page.</p> <pre data-bbox="993 403 1490 432"><<GetHeading(cur, cur, A, cur)>></pre> <p data-bbox="993 457 1490 541">Note: The heading in column A is shown for the dimension that corresponds to where the heading cell with the formula is located.</p> <pre data-bbox="993 558 1490 617"><<GetHeading(cur, cur, cur, " - ")>></pre> <p data-bbox="993 642 1490 764">Note: If used within a row text cell, the result is all of the headings in the current column separated by a " - ". This allows users to repeat column headings after a manual page break.</p>

*Use of the *current* keyword for the parameter “reference” is limited. The *current* keyword cannot be used in the *GetHeading* function when it is entered as a “custom heading” in a row or column because circular reference is created. You can use the *current* keyword as the third parameter within a cell that is part of a text row or column which is not in the heading area of the grid. For example, if you add the following formula in cell A3 where row 3 is a text row: `<<GetHeading(cur, cur, cur, 1)>>`, the result is the first heading of column A. You can use the *GetHeading* function within a heading cell. However, the *reference* parameter should not use the *current* keyword. For example, when you enter in the heading cell of column B `<<GetHeading(cur, cur, A, cur)>>`, the column A heading is returned.

Annotation

Annotation returns the requested information about an annotation within a grid object. You can define what information to return.

Syntax:

```
<<Annotation("GridName", Row, Col, Page, Attributes, Reference, Attachments)>>
```

```
<<Annotation("GridName", Row, Col, Page)>>
```

```
<<Annotation("GridName", Row, Col, Page, Attributes, Reference)>>
```

```
<<Annotation("GridName", Row, Col, Page, Attributes, Attachments)>>
```

```
<Annotation("GridName", AllRows, AllCols, Page, Attributes, Attachments)>>
```

where:

- *GridName* is the name of the grid containing the annotations. The “current” keyword is supported when *Attribute* is used in a grid's text cell or custom heading.
- *Row* is the row number on the grid. The “current” keyword is supported in when *Attribute* is used in a grid's text cell or custom heading.

Note: To select all rows, *row* can be replaced with the keyword `AllRows`.

- *Col* is the column reference (letter) on the grid. The “current” keyword is supported when *Attribute* is used in a grid's text cell or custom heading.

Note: To select all Columns, *Col* can be replaced with the keyword `AllCols`.

- *Page* is a numeric value representing the index of member combinations on all page dimensions of the grid. The first page dimension combination in a grid has an index of 1, the second page dimension combination has an index of 2, and so on.
- *Attributes* identify the information to be returned from the annotation. They may include the keyword **All** (default value) or a combination of one or more of the following keywords: `Title & Description & Author & Date & Category`. Attribute keywords can be arranged in any order of output. Each attribute must be separated with an ampersand (&).
- *Reference* is a keyword noting which annotations will be returned, where:
 - `All` (default) returns the initial annotation and all replies.
 - `Top <n>` returns the initial annotation and first <n> replies.
 - `Bottom <n>` returns the initial annotation and the last <n> replies. The initial annotation is always shown and the number <n> does not include the initial annotation.
 - `BottomOnly <n>` returns the last <n> replies. The initial annotation shown for 4 or less <n> replies; the initial annotation with 5 or more <n> replies is not shown.
 - `First` returns the initial annotation only.
 - `Replies` returns all replies. The initial annotation is not shown.
- *Attachments* is a boolean (`true/false`) value indicating whether attachments associated with the annotation should print with the report where:
 - `True` means that attachments will print at the end of the report.
 - `False` means that no attachments will print.

Examples:

```
<<Annotation("Grid1", cur, A, cur, All, All, true)>>
```

```
<<Annotation("Grid1", cur, A, cur, Title & Description & Author & Date  
& Category, Top 10, true)>>
```

```
<<Annotation("Grid10", 284, AB, cur, Title & Description & Date &  
Category, Bottom 10, false)>>
```

```
<<Annotation("Grid1", cur, A, cur, Title & Description & Author,  
BottomOnly 4, false)>>
```

```
<<Annotation("Grid Name", 1, A, 1, All)>>
```

```
<<Annotation("Grid Name", "AllRows", "AllCols", 1, All)>>
```

```
<<Annotation("Grid Name", 1, A, 1, All, First)>>
```

```
<<Annotation("Grid Name", 1, cur, cur, All, Replies)>>
```

```

<<Annotation("Grid Name", 1, cur, cur)>>
<<Annotation("Grid Name",1, cur, cur, Desc, Replies)>>
<<Annotation(cur, cur, A, cur, Desc)>>
<<Annotation(cur, 3, ABC, cur, title & Desc, true)>>
<<Annotation("folder1\folder2\Grid2", 3,A,cur, Title & Description,
true)>>
<<Annotation("current", 294, AB, cur, Title & Description, Top 19)>>
<<Annotation(current, 39, AB, cur, Title & Description & Author, Bottom
40)>>

```

Behavior of Annotations in Text Boxes

The behavior of annotations in grid text rows and text boxes when using the “cur” keyword is different.

- For text rows, you can use the “cur” keyword.
For example, <<Annotation("Grid1", 1, A, cur, All, All, true)>>
- For text boxes, the “cur” keyword does not return anything. As a result, you would enter <<Annotation("Grid1", 1, A, 1, All, All, true)>>

CellText

Subtopics

- [CellText Using AllRows/AllCols](#)
- [CellText Using MemberOverride](#)

CellText is a text function that retrieves the textual content from the following database connections:

The row, column, and page arguments apply to the intersection in a grid that contains the linked text or note.

Syntax:

```
<<CellText("GridName", Row, Column, Page)>>
```

CellText can also provide an override of the grid’s “natural” point of view (POV) to a member selection when a cell does not exist in a grid.

Syntax:

```
<<CellText("GridName", Row, Column, Page, MembeOverride)>>
```

See [“CellText Using MemberOverride” on page 112](#) for details.

Argument	Description
<i>GridName</i>	Name of a grid. This value must be enclosed in quotation marks
<i>Row</i>	Numeric value representing the row number of the grid. The first row in a grid has an index of 1, the second row has an index of 2, and so on.
<i>Column</i>	Letter that represents the column of the grid. The first column in a grid has an index of A, the second column has an index of B, and so on.
<i>Page</i>	Numeric value representing the page dimension index of the grid. The first page dimension in a grid has an index of 1, the second page dimension has an index of 2, and so on.
<i>MemberOverride</i>	String of dimensions and corresponding member selections that override the <code>CellText</code> POV. See “CellText Using MemberOverride” on page 112 .
<i>AllRows/AllCols</i>	Keywords that allow for an all row or column reference. See “CellText Using AllRows/AllCols” on page 111 .

Note: If the row or column parameters refer to a segment that expands, the top-left expanded cell is used. You can refer to any expanded cell using the “range” notation.

Example:

Create a report with linked text in the cell that is in row 21, column B of a grid and that is on the first page dimension.

```
<<CellText ("mygrid" , 21 , B , 1) >>
```

Tip: To view result-time rows and columns, select **View**, then **Row & Column Headers in Web Preview**, and then preview the report by clicking **Web Preview**.

Note: `CellText` only supports the `Current` keyword when used in a grid cell; not in a text box object, header, or footer. See [“Using the Current/Cur Keyword in a Text Function” on page 104](#).

CellText Using AllRows/AllCols

Table 11 CellText Examples Using AllRows/AllCols

Example	Description
<code><<CellText ("Grid1" , 3 , AllCols , 1) >></code>	Includes cell text that occurs on any column of row 3, page 1.
<code><<CellText ("Grid1" , Curr , Allcols , 1) >></code>	Includes cell text that occurs on any column of the current row (“Curr”), page 1.
<code><<CellText ("Grid1" , Curr , Allcols , Curr) >></code>	Includes cell text that occurs on any column of the current row (Curr), current page.
<code><<CellText ("Grid1" , AllRows , C , 1) >></code>	Includes cell text that occurs on any row of column C, page 1.

Example	Description
<<CellText("Grid1",AllRows,Allcols,2)>>	Includes cell text that occurs on any cell on page 2. "Any cell" meaning any row/column since AllRows/AllCols is specified for both row and column.
<<CellText("Grid1", AllRows, Allcols, Curr)>>	Includes cell text that occurs on any cell on the current page.

CellText Using MemberOverride

MemberOverride can consist of one or more dimensions. It uses the following format:

DimensionName = *MemberName*, *DimensionName* = *MemberName*

where:

DimensionName is the name of an override dimension.

MemberName is the name of the override member for the dimension.

Follow these guidelines when you specify member overrides:

- If you specify two or more member overrides, separate them with a comma (,) or a semicolon (;) and enclose each dimension and member name in quotation marks (" "), For example:

```
"Entity"="Acme", "Period"="Q1"
```

- If you specify only one member override, do not include the separator character.
- Enclose dimension names and member names in quotation marks (" ") if they contain any of the following characters: ;, = () < >.
- For a single member override, the parameter may be enclosed in quotation marks (" "). For example, CellText is properly evaluated if you use the following member override statement:

```
"Scenario=Budget"
```

Table 12 CellText Examples Using MemberOverride

Example	Description
<<CellText(cur, cur, A, cur)>>	Existing syntax with no dimension override
<<CellText(cur, cur, A, cur, Scenario=Budget)>>	New syntax with one dimension override
<<CellText("Grid Name", 1, A, current, Value = "Entity Currency")>>	One dimension override using quotation marks in the member name
<<CellText("Grid Name", 1, A, cur, Value = Entity Currency)>>	No quotation marks in the member name that contains a space
<<CellText("Grid Name", 1, A, cur, "Value=\$USD" = Entity Currency)>>	Quotation marks surrounding dimension name that contains an equal sign
<<CellText(Grid1, 1, cur, cur, Value = Entity Currency, Scenario=Actual)>>	Two-dimension override that uses a comma separator

Example	Description
<<CellText(Grid1, 1, cur, cur, Value = Entity Currency; Scenario=Actual)>>	Two-dimension override that uses a semicolon separator
<<CellText("Current", 34, BB, cur, "Value"= Entity Currency, Scenario=Actual; Period = Qtr3)>>	Three-dimension override that uses comma and semicolon separators
<<CellText(cur, 1(3), A(B), cur, 300 = ABC , "Americas, Value=(in \$USD);" = "Entity Currency (USD)" , Scenario=Actual)>>	Four-dimension override

Date

Date is a text function that returns the date and time that an online report is populated with data, or the date and time that a snapshot report is saved. The date and time is retrieved from the report server and corresponds to the country in which the report server is located. You can use Date in a text box, text cell, or a row or column heading.

Syntax:

```
<<Date("format", "TimeZoneId")>>
```

Argument	Description
<i>format</i>	Defaults to the user preference for date/time. Valid values are those accepted by Java SimpleDateFormat, or "user" or empty ("user"). Date and time <i>format</i> characters are case-sensitive.
<i>TimeZoneId</i>	Defaults to user preferences for the time zone. Valid values are those accepted by Java TimeZone.getTimeZone(), or the server time zone. For example, for TimeZone.getTimeZone() to specify the east coast, enter: "America/New_York". When using <i>TimeZoneID</i> , enclose the <i>format</i> parameter within quotation marks. This applies only when <i>TimeZoneId</i> is indicated. You do not need to update reports prior to the <i>TimeZoneId</i> enhancement.

Table 13 Date and Time Format Characters

Format Characters	Meaning	Data Type	Example
G	era designator	text	AD
M	month in year	text or number	If the month is July: M displays: 7 MM displays: 07 MMM displays: Jul MMMM displays: July
d	day in month	number	10
h	hour in am/pm (1-12)	number	11
H	hour in day (0-23)	number	22
m	minute in hour	number	30
s	second in minute	number	25

Format Characters	Meaning	Data Type	Example
S	millisecond	number	978
E	day in week	text	E: Tue EE: Tuesday
D	day in year	number	189
F	day of week in month	number	<<Date("dd-MMM-yy 'is the' F 'th ' E 'of' 'MMM")>> displays: 24-Dec-02 is the 4th Tue of Dec
w	week in year	number	27
W	week in month	number	2
a	am/pm marker	text	PM
k	hour in day (1-24)	number	23
K	hour in am/pm (0-11)	number	If the time is 3:37PM: 3
z	time zone	text	Pacific Standard Time
y	year	number	2002,02
'	literal text	text	<<Date(" D'the day of 'yyyy")>> displays:105th day of 2002
''	two single quotes to produce one single quote	text	<<Date("k:mm 'O'Clock' a")>> displays: 6:15 O'Clock PM Note: Use 2 single quotation marks to produce a single quote or an apostrophe as in O'Clock (O''Clock).

Considerations when using *format* characters:

- The number or count of *format* characters determines the format:
 - For text, less than four pattern letters means use a short or abbreviated form, if it exists.
 - Four or more pattern letters means use full form.
 - M or MM means use the month as a number.
 - MMM means use the three-letter abbreviation for the month.
 - MMMM means use the full name of the month.

For example, if the date is April 24:

<<Date("dd-M-yy")>> displays 24-4-02

<<Date("dd-MM-yy")>> displays 24-04-02

<<Date("dd-MMM-yy")>> displays 24-Apr-02

<<Date("dd-MMMM-yy")>> displays 24-April-02

- For numbers, the number of pattern letters is the minimum number of digits. Shorter numbers are padded with zeros. Year is a special case. If you use “yy”, the last two digits of the year are displayed, but if you use “yyyy,” the four-digit year is displayed.

For example, if the month is February:

```
<<Date("MM-yyyy")>> displays 02-2002
```

- All upper and lower case alphabet characters are considered patterns, regardless of whether they are designated *format* characters.
- Enclose literal text within single quotation marks. To use a single quotation mark in literal text, enclose the text within single quotes and use two single quotes for the apostrophe.

For example, the format to print 4 o'clock is:

```
<<Date("hh 'o' 'clock' ")>>
```

Examples:

Date Format	Result
<<Date("d/M/yy")>>	23/3/02
<<Date("d-MMM-yy")>>	23-Mar-02
<<Date("EEEE, MMMM dd, yyyy")>>	Tuesday, March 23, 2002
<<Date("h:mm:ss a")>>	12:52:05 PM
<<Date("h:mm:ss a zzzz")>>	12:52:05 PM Eastern Standard Time
<<Date("EEEE, MMMM dd, yyyy G 'time:' h:mm:ss a zzzz")>>	Tuesday, March 23, 2002 AD time: 12:52:05 PM Eastern Standard Time
<<Date("hh 'o' 'clock' a, zzzz")>>	12:00 PM, Eastern Standard Time Note: You must use two single quotation marks to produce one single quotation mark in the text of your formatted result.

DataSource

DataSource is a text function that returns the application name, database, or alias table name of a grid. Use DataSource in a text box, text cell, or a row or column heading.

Syntax:

```
<<DataSource("GridName.Axis[ID]", InfoType)>>
```

Argument	Description
GridName	(Required) Any grid in a report, enclosed in quotes.

Argument	Description
<i>Axis[]</i>	<p>(Optional) A pointer to a design segment of the grid. Axis can be one of the following keywords: row, col, or column, and it is not case-sensitive.</p> <p>For example,</p> <pre><<DataSource("Grid1.Row[1]",DB)>></pre> <p>When <i>Axis</i> is not provided, the grid's default information is returned. Use <i>Axis</i> when a grid has multiple database connections.</p>
<i>ID</i>	<p>(Required if <i>Axis</i> is used) Indicates the design-time, row number, or column letter from which the database connection information is retrieved. If <i>ID</i> points to an invalid or nonexistent row or column ID, the grid default settings are used. In addition, if a grid does not have a secondary database connection, the grid default settings are used.</p> <p>For example,</p> <pre><<DataSource("Grid1.Col[A]",DB)>></pre>
<i>InfoType</i>	<p>(Required) One of the following keywords:</p> <ul style="list-style-type: none"> ● <i>App</i> returns the application name. ● <i>DB</i> returns the database name. ● <i>Alias</i> returns the alias table name. To return the alias, the grid must have the alias table enabled. By default, the alias table is disabled. ● <i>Name</i> returns the database connection name associated with the specified grid. <p>Note: "Server" is not supported as an argument for <i>InfoType</i>. for Planning data sources.</p>

Note: All arguments are case insensitive.

Example 1:

Insert data source information in your report:

```
<<DataSource("Grid1", App)>>
```

Example 2:

Show the database name of the data source associated with the heading where the function was entered:

```
<<DataSource(cur, DB)>>
```

Note: *DataSource* supports the *current* keyword. See [“Using the Current/Cur Keyword in a Text Function” on page 104.](#)

Footnote

The *Footnote* text function, used in Text Objects only, returns the requested information about footnotes within a given report object. *Footnote* prints information about footnotes that occur in the object specified, or in all the objects in the report if the first parameter is set to "all". When a grid is referenced, all the footnotes on the grid are printed. The partial POV footnotes print

first, followed by any full POV footnotes. The full POV footnotes print in order based on the location of the cell from top left to bottom right, page by page. More than one syntax is supported.

Syntax:

```
Footnote("ObjectName", Attributes, Reference, Attachments)
```

```
Footnote("ObjectName")
```

```
Footnote("ObjectName", Attributes)
```

```
Footnote("ObjectName", Attributes, Reference)
```

```
Footnote("ObjectName", Attributes, Attachments)
```

Argument	Description
<i>ObjectName</i>	Name of the report object containing the footnotes. If "all", the function returns all available footnotes from all objects in the report. The <i>ObjectName</i> parameter is updated when a grid object is renamed.
<i>Attributes</i>	Information returned from the footnote. This may include the following keywords: <ul style="list-style-type: none"> ● All (default value) ● A combination of one or more of these keywords; Title, Description (Desc), Author, Date, Category <p>The <i>Attributes</i> keywords may be arranged in the preferred order of output. Separate each attribute with an ampersand (&).</p>
<i>Reference</i>	Keyword denoting which annotations will be returned where: <ul style="list-style-type: none"> ● All (default value)—The initial annotation and all replies. ● Top <n>—The initial annotation and first replies. (The initial annotation is always shown, and the number does not include the initial annotation.) ● Bottom—The initial annotation and the last number of replies specified in Bottom <n>. (The initial annotation is always shown, and the number does not include the initial annotation.) ● BottomOnly <n>—The last replies. The initial annotation is not shown unless there are less replies than the BottomOnly <n> number. For example, BottomOnly 5 for an annotation with five replies does not show the initial annotation, but for an annotation with four replies, the initial annotation is shown. ● First—The initial annotation only. ● Replies— All the replies except the initial annotation.
<i>Attachments</i>	Boolean (true/false) value indicating whether any attachments should be printed with the footnote. <ul style="list-style-type: none"> ● True—Attachments print along with the report, at the end. ● False (default value)—Attachments do not print.

Examples:

```
<<Footnote("Grid1", All, All, true)>>
```

```
<<Footnote(All, All, All, true)>>
```

```
<<Footnote(All, All, Bottom 2, true)>>
```

```
<<Footnote(All, All, BottomOnly 20, true)>>
```

```
<<Footnote(All, All, First, true)>>
```

```

<<Footnote(All, All, Replies, true)>>
<<Footnote("Grid1", Title & Description & Author & Date & Category,
All, true)>>
<<Footnote("Text1", Title & Description & Date & Category, Top 10,
false)>>
<<Footnote("Image1", Title & Description & Author, Bottom 4, false)>>
<<Footnote("Text1", Title&Description&Author, BottomOnly 12)>>
<<Footnote("Image1", Title&Description&Author, First)>>
<<Footnote("Grid2", Title&Description&Author, Replies)>>
<<Footnote("Grid Name")>>
<<Footnote(all, All)>>
<<Footnote(cur, Description)>>
<<Footnote(cur, title & Description, true)>>
<<Footnote(cur, title & Author & Description, false)>>
<<Footnote("folder1\folder 2\Grid1", title & Description&Category,
Top20)>>
<<Footnote("folder1\folder 2\Grid1", title & Description&Category,
Bottom 2)>>
<<Footnote("folder1\folder 2\Chart2", title & Description, Top 20,
False)>>
<<Footnote("current", title & Description, All)>>
<<Footnote("current", title & Description, first)>>
<<Footnote(All, Title&Description&author, Top 40, true)>>

```

GetCell

GetCell is a text function that returns a data value from a grid.

Note: When referencing a cell containing Planning or Oracle Hyperion Financial Management data, GetCell returns a string value instead of a number.

Syntax:

```
<<GetCell("GridName", Row, Column, Page)>>
```

Argument	Description
<i>GridName</i>	Name of a grid or the Current keyword.

Argument	Description
<i>Row</i>	Numeric value representing the row number of the grid. The first row in a grid has an index of one, the second row has an index of two, and so on. The <code>Current</code> keyword can also be used.
<i>Column</i>	Letter representing the column of the grid. The first column in a grid has an index of A, the second column has an index of B, and so on.
<i>Page</i>	Numeric value representing the page dimension index of the grid. The first page dimension in a grid has an index of one, the second page dimension has an index of two, and so on.

Considerations when using `GetCell`:

- If the row or column parameters refer to a segment that expands, the top-left expanded cell is used. You can refer to any expanded cell using the “range” notation.
- `GetCell` only supports the `Current` keyword when used in a grid cell; not in a text box object, header, or footer. The `Current` keyword can be used in any of the parameters to make the function more dynamic. See [“Using the Current/Cur Keyword in a Text Function” on page 104](#).

Example:

Set up a grid to show the net income for several quarters, and display this value in a text box that contains an executive summary of the report.

The grand total for the period was `<<GetCell("mygrid",21,B,1)>>`

The report is created with the value in the cell that is in row 21, column B of a grid, and is on the first page dimension.

GetHeading

`GetHeading` is a text function that obtains text headings from a specified grid.

Syntax:

`<<GetHeading("GridName", Page, Reference, Offset)>>`

`<<GetHeading("GridName", Page, Reference, "Delimiter")>>`

Argument	Description
<i>GridName</i>	Name of a grid containing the heading to extract. The <code>Current</code> keyword can also be used.
<i>Reference</i>	View time reference of the heading to obtain. The <code>Current</code> keyword can also be used. See “Row, Column, or Cell Reference Arguments” on page 87 .
<i>Offset</i>	Number that represents the dimension in a row or column heading with more than one dimension. A numeric offset returns a single dimension heading. For example, in a row heading with three dimensions, you can specify (starting from left to right) “1” to return the first dimension, “2” to return the second dimension, and “3” to return the third dimension.

Argument	Description
"Delimiter"	Quoted string that separates all headings from the row or column. If you provide a delimiter, you obtain all headings in the reference, separated by the provided delimiter.

Examples:

Use `GetHeading` to return the following column and row headings.

		A (A)	A (B)	A (C)
		Actual	Actual	Actual
		Jan	Feb	Mar
1 (1)	Georgia	112	67	73
1 (2)	East	5,864	3,322	3,789
1 (3)	Market	15,904	9,277	10,640

Use the syntax below to return the data in the previous sample grid:

GetHeading Example	Gets or Returns
<code>GetHeading("Grid1",1,A,2)</code>	Jan
<code>GetHeading("Grid1",1,A(B),2)</code>	Feb
<code>GetHeading("Grid1",1,A(C),"-")</code>	Actual - Mar
<code>GetHeading("Grid1",1,1,1)</code>	Georgia
<code>GetHeading("Grid1",1,1(2),1)</code>	East
<code>GetHeading("Grid1",1,1(2),2)</code>	<error> (it is an invalid reference)
<code>GetHeading("Grid1",1,1(3),"-")</code>	Market

Note: `GetHeading` only supports the `Current` keyword when used in a grid cell; not in a text box object, header, or footer. See [“Using the Current/Cur Keyword in a Text Function” on page 104.](#)

GridDimension

`GridDimension` is a text function that returns the name of a dimension on a grid's page, column, or row axis.

Syntax:

```
<<GridDimension("GridName", "Axis", index)>>
```

Argument	Description
<i>GridName</i>	Name of a grid in the report enclosed in quotation marks. The <code>Current</code> keyword can also be used.
<i>Axis</i>	Grid axis where the dimension is placed: page, row, or column. Keywords must be used. The <code>Current</code> keyword can also be used.
<i>index</i>	Numeric value representing the offset or index of the dimension on the named axis. The first dimension on an axis has an index of one, the second dimension has an index of two, and so on. You can use the <code>Current</code> keyword on a heading cell only (page, row, or column).

Example:

A shared grid, named *mygrid*, has several dimensions on each row, column, and page axes, and displays the name of the first dimension on the row axis in a text box. Because the dimension may change in the future, you use the `GridDimension` function in the text box. Type the following in the text box:

This report is based on the `<<GridDimension("mygrid","Row",1)>>` dimension.

Note: See [“Using the Current/Cur Keyword in a Text Function” on page 104.](#)

ListofCellDocuments

`ListofCellDocuments` is a text function that returns a list of all cell documents and their file attributes selected to be retrieved from cells within a report. You can use this function in a text box, text cell, or text cell heading.

Note: `ListofCellDocuments` supports the `Current` keyword. See [“Using the Current/Cur Keyword in a Text Function” on page 104.](#)

Syntax:

`ListofCellDocuments ("GridName", FileName, Description)`

Argument	Description
<i>GridName</i>	Name of a grid in the report.
<i>FileName</i>	Name of the cell document.
<i>Description</i>	Description of each cell document.

Example:

Two grids in a report are selected to print cell document information. In `Grid1`, the file name and description of the extracted cell documents are requested. In `Grid2` only the file name is requested.

`<<ListOfCellDocuments ("Grid1", FileName, Description)>>`

```
<<ListOfCellDocuments ("Grid2",FileName)>>
```

The resulting list might look similar to the following, where Grid1 has two attached documents, and Grid2 has one attached document:

Attached Documents:

April Variance High Volume
April Variance Low Volume
May Variance

MemberAlias

Note: This function is available only with Essbase Cloud or Planning details as the database connection.

MemberAlias is a text function. Based on the syntax used, it returns the alias of the member assigned to a dimension in the row, column, page, or Point of View (POV).

- The following syntax returns the alias of a row heading, column heading, or page. The syntax can be deployed from any heading cell, text cell, or text box object and can reference any grid in a current report.

```
<<MemberAlias ("GridName", Row/Col/Page, "DimName")>>
```

- The following syntax returns the alias in a row heading, column heading, or page. The syntax can be deployed from any row, column, or page heading in a grid that is designated as a custom heading. To designate a custom heading, click the page, row heading, or column heading, and then select Custom Heading on the property sheet. You can add the function in the space provided below the custom heading.

```
<<MemberAlias ("DimName")>>
```

- The following syntax returns the alias of a grid or user POV. The syntax can be deployed from any text cell or text box object and can reference any grid and corresponding POV in a current report.

```
<<MemberAlias ("GridName","POV DimName")>>
```

Argument	Description
<i>GridName</i>	Name of the grid
<i>Row/Col/Page</i>	Numeric value representing the index of member combinations on all page dimensions of the grid. The first page dimension combination in a grid has an index of 1, the second page dimension combination has an index of 2, and so on.
<i>DimName</i>	Name of a dimension in the grid row, column, or page.
<i>POV DimName</i>	Name of the dimension on the grid or user POV.

Examples:

Yield the alias Diet Root Beer, which is the alias that is assigned to the Product dimension in Grid1, column A:

```
<<MemberAlias("Grid1", A, "Product")>>
```

```
<<MemberAlias(current, A, "Product")>>
```

Yield the alias for the Year dimension. The syntax is placed on the grid's custom heading:

```
<<MemberAlias("Year")>>
```

```
<<MemberAlias(current)>>
```

Yield the alias for the Scenario dimension for the POV associated with Grid1:

```
<<MemberAlias("Grid1", "Scenario")>>
```

```
<<MemberAlias(current, "Scenario")>>
```

Note: MemberAlias only supports the Current keyword when used in a grid cell or in a text box object when specifying the Page dimension parameter. Other parameters (Grid, Row, Column) are not supported in a text box object, header, or footer. See [“Using the Current/ Cur Keyword in a Text Function” on page 104.](#)

MemberName

dMemberName is a text function. Based on the syntax used, it returns the name of the member assigned to a dimension in the row, column, page, or Point of View (POV).

- The following syntax returns the name of a row heading, column heading, or page. The syntax can be deployed from any heading cell, text cell or text box object and can reference any grid in a current report.

```
<<MemberName("GridName", Row/Col/Page, "DimName")>>
```

- The following syntax returns the name in a row heading, column heading, or page. The syntax can be deployed from any row, column, or page heading in a grid that is designated as a custom heading. To designate a custom heading, click the page, row heading, or column heading, and then select Custom Heading on the property sheet. You can add the function in the space provided below the custom heading.

```
<<MemberName("DimName")>>
```

- The following syntax returns the name of a grid or user POV. The syntax can be deployed from any text cell or text box object and can reference any grid and corresponding POV in a current report.

```
<<MemberName("GridName", "POV DimName")>>
```

Argument	Description
GridName	Name of the grid.

Argument	Description
<i>Row/Col/Page</i>	Numeric identifier of the row, the alphabetical identifier of the column, or the numeric identifier of the page.
<i>DimName</i>	Name of a dimension in the grid row, column, or page.
<i>POV DimName</i>	Name of the dimension on the grid or user POV.

Examples:

Return the name assigned to the Product dimension in Grid1, column A:

```
<<MemberName("Grid1", A, "Product")>>
```

```
<<MemberName(current, A, "Product")>>
```

Return the name for the Year dimension. The syntax is placed on the grid's custom heading:

```
<<MemberName("Year")>>
```

Return the name for the Scenario dimension for the POV associated with Grid1:

```
<<MemberName("Grid1", "Scenario")>>
```

```
<<MemberName(current, "Scenario")>>
```

Note: MemberName only supports the Current keyword when used in a grid cell or in a text box object when specifying the Page dimension parameter. Other parameters (Grid, Row, Column) are not supported in a text box object, header, or footer. See [“Using the Current/Cur Keyword in a Text Function” on page 104](#).

MemberProperty

Note: MemberProperty is applicable to Planning details and Essbase Cloud database connections.

MemberProperty is a text function. Based on the syntax used, it returns the member property value of a dimension in the row, column, page, or Point of View (POV). You can use this function to display a member property in a text row or column.

- The following syntax returns the member property of a row heading, column heading, or page. The syntax can be deployed from any heading cell, text cell or text box object and can reference any grid in a current report.

```
<<MemberProperty("GridName", Row/Col/Page, DimName, Property)>>
```

- The following syntax returns the member property in a row heading, column heading, or page. The syntax can be deployed from any row, column, or page heading in a grid that is designated as a custom heading. To designate a custom heading, click the page, row heading, or column heading, then select Custom Heading on the property sheet. You can add the function in the space provided below the custom heading.

```
<<MemberProperty("DimName", Property)>>
```

- The following syntax returns the member property of a grid or user POV. The syntax can be deployed from any text cell or text box object and can reference any grid and corresponding POV in a current report.

```
<<MemberProperty("GridName", POV DimName, Property)>>
```

Argument	Description
<i>GridName</i>	Name of the grid.
<i>Row/Col/Page</i>	Numeric identifier of the row, the alphabetical identifier of the column, or the numeric identifier of the page.
<i>POV DimName</i>	Name of the dimension on the grid or user POV.
<i>Property</i>	Any custom property (attribute dimensions).

Examples:

Return the member property assigned to the Product dimension in Grid1, column A:

```
<<MemberProperty("Grid1", A, Product, Pkg Type)>>
```

```
<<MemberProperty(current, A, Product, Pkg Type)>>
```

Return the member property for the Market dimension. The syntax is placed on the grid's custom heading:

```
<<MemberProperty("Market", Population)>>
```

```
<<MemberProperty(current, Population)>>
```

Return the member property for the Market dimension for the POV associated with Grid1:

```
<<MemberProperty("Grid1", Market, CurrencyCategory)>>
```

```
<<MemberProperty(current, Market, CurrencyCategory)>>
```

Retrieve the Member Alias and Ounces property of a product in a custom heading. (To do this, select the heading, then on the Heading Row Properties sheet, select the Custom Heading option, and then click Functions).

```
<<MemberAlias(current, current, Product)>>:
```

```
<<MemberProperty(current, current, Product, Ounces)>>
```

The report output would resemble the following:

	Qtr 1
Cola : Ounces_12	5,096
Diet Cola : Ounces_12	1,359
Caffeine Free Cola : Ounces_16	593

Note: MemberProperty only supports the Current keyword when used in a grid cell, not in a text box object, header, or footer. See [“Using the Current/Cur Keyword in a Text Function” on page 104.](#)

Page

Page is a text function that returns the current page number of a printed report. Use this function in a text object.

Syntax:

```
<<Page()>>
```

Example:

If the current page is eight, yield: Page 8.

```
Page<<PAGE()>>
```

Note: Page works for printed reports only. Online reports display `[[PageCount()]]` for the page number. The page axis on a grid is different from the page number in a printed report.

PageCount

PageCount is a text function that returns the total number of pages in a printed report. Use this function in a text object.

Syntax:

```
<<PageCount()>>
```

Example:

If the current page count is 6, yields: “The total number of pages is: 6”.

```
The total number of pages is: <<PageCount()>>
```

Note: PageCount works for printed reports only. Online reports display `[[PageCount()]]` for the page number.

PlanningAnnotations

PlanningAnnotations is a text function that retrieves the critical notes or comments associated with a cell’s Planning Unit from the Planning database connection. Planning Units are a combination of Scenario, Version, and Entity and are derived in Financial Reporting through cell references in a grid.

Syntax:

```
<<PlanningAnnotations("GridName", Row, Column, Page, Attributes,  
Range)>>
```

```
<<PlanningAnnotations("GridName", Row, Column, Page, Attributes)>>
```

Argument	Description
<i>GridName</i>	(Required) Name of a grid.
<i>Row</i>	(Required) Numeric value representing the row number of the grid. The first row in a grid has an index of 1, the second row has an index of 2, and so on.
<i>Column</i>	(Required) Letter that represents the column of the grid. The first column in a grid has an index of A, the second column has an index of B, and so on.
<i>Page</i>	(Required) Numeric value representing the index of member combinations on all page dimensions of the grid. The first page dimension combination in a grid has an index of 1, the second page dimension combination has an index of 2, and so on.
<i>Attributes</i>	Any of the following values: All, Title, Author, Date, Text, PlanningUnit. (PlanningUnit is the combination of scenario, version, and entity and is derived from the <i>GridName</i> , <i>Row</i> , <i>Column</i> , and <i>Page</i> .) Arrange attributes in the preferred order of output, and separate each attribute with an ampersand (&).
<i>Range</i>	Use the keywords All, Top, or Bottom. to select the number of annotations from the top or bottom of an output, or all annotations. <ul style="list-style-type: none"> ● All returns all annotations ● Top 5 returns the first five annotations ● Bottom 10 returns the last ten annotations ● Bottom 1 returns the last annotation

Example 1:

“All” is assumed for the *Range* parameter.

```
<<PlanningAnnotations("Grid Name", 1,a,1,All, All)>>
<<PlanningAnnotations("Grid Name", 100,AB,10,All, All)>>
<<PlanningAnnotations(cur, cur,a,cur,Text & Title & Author, Top 5)>>
<<PlanningAnnotations(cur, 315, AB, 255, Text&Title&Author, Top 5)>>
<<PlanningAnnotations(cur, cur A, Cur, PlanningUnit, Top 5)>>
<<PlanningAnnotations("cur, 123, ABC, 101, PlanningUnit, Top5)>>
<<PlanningAnnotations("Grid1, 1, current, cur, Title & Text, Bottom 10)>>
<<PlanningAnnotations("Grid1, 105, ABC, cur, Title & Text, Bottom 10)>>
<<PlanningAnnotations("Current", 34, BB, cur, "All", Top 40)>>
<<PlanningAnnotations(cur, cur, A cur, "Text & Title & Author", Top 5)>>
<<PlanningAnnotations(cur, cur, A, cur, "Text&Title&Author, "bottom15")>>
<<PlanningAnnotations(cur, 1(3), A(B), cur, Title&Author&Date&Text, "All")>>
<<PlanningAnnotations(cur, 1(3), A(B), cur, "Title&Author&Date&Text, "All")>>
```

Example 2:

Retrieve annotation text in row 1, column A, of a grid on the current page. Display the bottom three annotations and all attributes associated with the annotation (title, author, date, text, and planning unit).

Use this syntax in a text cell:

```
<<PlanningAnnotations(Current, 1, A, Current, All, Bottom 3)
```

Annotations are returned chronologically in descending order by date, with the most recent annotations on top and the oldest annotation on the bottom. The resulting annotation text resembles the following:

Title: Status - Under Review
Author: John Smith
Date: Mar 25, 2003 10:32:49 AM
Planning Unit: Budget, 1st Draft, East
Text: Please review and approve

Title: Status - Not Signed Off
Author: Mary Brown
Date: Mar 21, 2003 2:59:11 PM
Planning Unit: Budget, 1st Draft, West
Text: Sorry, Try Again

Title: Status - Under Review
Author: Admin
Date: Mar 21, 2003 2:54:16 PM
Planning Unit: Budget, 1st Draft, South
Text: Please review and approve budget for 1st draft

Note: PlanningAnnotations only supports the Current keyword when used in a grid cell; not in a text box object, header, or footer. See [“Using the Current/Cur Keyword in a Text Function” on page 104.](#)

ReportAuthor

ReportAuthor is a text function that returns the user name of the person who created the report. Use this function in a text box, text cell, row, or column heading.

Syntax:

<<ReportAuthor ()>>

Example:

Return the name of the report author:

<<ReportAuthor ()>>

ReportCreated

ReportCreated is a text function that returns the date a report was created. Use this function in a text box, text cell, row, or column heading.

Syntax:

```
<<ReportCreated ("Format", "TimeZoneId") >>
```

Argument	Description
<i>format</i>	Characters, enclosed in quotation marks, that define the format of the date and time. <i>Format</i> defaults to the user preference for date /time. Valid values are those accepted by Java <code>SimpleDateFormat</code> , "user", or empty ("user"). For a detailed description of date and time formats, see Table 13 on page 113 .
<i>TimeZoneId</i>	Defaults to user preferences for the time zone. Valid values are those accepted by Java <code>TimeZone.getTimeZone()</code> , or the server time zone. For example, for <code>TimeZone.getTimeZone()</code> to specify the east coast, enter: <code>America/New_York</code> .

Note: The date and time format is case-sensitive. ReportCreated returns a value only after the report is saved.

Example:

Insert the report creation date in the body of a report:

```
<<ReportName () >> - Created on <<ReportCreated ("d-MM-yy") >>
```

ReportDesc

ReportDesc is a text function that returns the description of the current report. Use this function in a text box, text cell, row, or column heading.

Syntax:

```
<<ReportDesc () >>
```

Example:

Insert a report description in the body of a report:

```
<<ReportDesc () >>
```

Note: You can define a report description when you save the report. Afterward, you can change the description in the repository or when you save the report again with the Save As command.

ReportFolder

ReportFolder is a text function that returns the path of the folder where the report is located. Use this function in a text box, text cell, row, or column heading.

Syntax:

```
<<Reportfolder ( )>>
```

Example:

Insert the path of the folder where the report is located:

```
<<Reportfolder ( )>>
```

Considerations when using ReportFolder:

- The report must be located in a folder other than the root folder. The root folder is not considered a report folder and returns an empty string.
- The report must be saved to a folder in order to return the correct string. An unsaved report returns an empty string.
- If you open a report containing a ReportFolder function, save it to a new folder. When you use Print Preview to preview the report, the previous location of the folder is displayed. This is by design for performance reasons. To reevaluate the report, modify a grid or report object, then use Print Preview to see the new folder location.

ReportModified

ReportModified is a text function that returns the date the current report was last modified. Use this function in a text box, text cell, row, or column heading.

Syntax:

```
<<ReportModified("format", "TimeZoneId")>>
```

Argument	Description
<i>format</i>	Characters, enclosed in quotation marks, that define the format of the date and time. <i>Format</i> defaults to the user preference for date /time. Valid values are those accepted by Java SimpleDateFormat, "user", or empty ("user"). For a detailed description of date and time formats, see Table 13 on page 113 .
<i>TimeZoneId</i>	Defaults to user preferences for the time zone. Valid values are those accepted by Java TimeZone.getTimeZone(), or the server time zone. For example, for TimeZone.getTimeZone() to specify the east coast, enter: America/New_York.

Note: The date and time format is case-sensitive. ReportModified returns a value only after the report is saved.

Example:

Insert the date Jan 19, 2013. (This is the date that the report was last modified.)

```
Report Modified: <<ReportModified("MMM dd, yyyy")>>
```

ReportModifiedBy

ReportModifiedBy is a text function that returns the user name of the last user to save the report. Use this function in a text box, text cell, row, or column heading.

Syntax:

```
<<ReportModifiedBy()>>
```

Example:

Insert the name of the last user to save the report:

```
<<ReportModifiedBy()>>
```

ReportName

ReportName is a text function that returns the name of the current report. Use this function in a text box, text cell, row, or column heading.

Syntax:

```
<<ReportName()>>
```

Example:

Insert the report name:

```
<<ReportName()>>
```

ReportRunBy

ReportRunBy is a text function that returns the user name of the user who is running the report. Use this function in a text box, text cell, row, or column heading.

Syntax:

```
<<ReportRunBy()>>
```

Example:

Insert the name of the user who is running the report:

```
<<ReportRunBy()>>
```

RetrieveValue

RetrieveValue functions in existing reports are no longer supported and must be manually updated to use the GetCell function.

Conditional Functions

Subtopics

- [IFThen, If](#)
- [Conditional Operators](#)
- [Complex Conditions](#)

IFThen, If

IfThen is a conditional function that returns a value when the condition equals True, and another value when the condition equals False.

Syntax:

`IfThen(Condition, TrueParameter, FalseParameter)`

- *Condition* is a logical expression that evaluates to true or false. Full conditional logic can be used as well as complex Boolean operators (And, Not, and Or). A *condition* can also test for #missing and #error values.
- *TrueParameter* and *FalseParameter* are expressions that are evaluated based on the outcome of the condition.

Conditional Operators

When using conditional operators, consider the following:

- *Expression* can be any valid formula expression. The expression can be any combination of a constant (integer or real number), a reference, or another function.
- *Reference* can be any valid reference; thus the IFNN reference property can be utilized as part of the reference.
- *Condition* can be any valid condition applied to the complex conditions And, Not, and Or. These operators can have embedded conditions. (And, Not, and Or operators require surrounding parentheses.)
- When any *expression* within the condition returns an #error or #missing value, the If function returns #missing or #error. This does not apply when you use the IsMissing, IsError, or IsNonNumeric conditions.

Table 14 Conditional Operators

Conditional Operator	Syntax	Logic
Equal To	<i>expression = expression</i>	Tests if the left expression is equal to the right expression. Note: If rounding is required, use the <code>Round</code> function. Example: <code>1 = 4</code> Returns false
Greater Than	<i>expression > expression</i>	Tests if the left expression is greater than the right expression. Example: <code>1 > 4</code> Returns false
Greater Than or Equal To	<i>expression >= expression</i>	Tests if the left expression is greater than or equal to the right expression. Note: The correct syntax is ">=". The syntax "=>" is not supported. Example: <code>1 >= 4</code> Returns false
Less Than	<i>expression < expression</i>	Tests if the left expression is less than the right expression. Example: <code>1 < 4</code> Returns true
Less Than or Equal To	<i>expression <= expression</i>	Tests if the left expression is less than or equal to the right expression. Note: The correct syntax is "<=". The syntax "=<" is not supported. Example: <code>1 <= 4</code> Returns true
Not Equal To	<i>expression <> expression</i> <i>expression != expression</i>	Tests if the left expression is not equal to the right expression. Note: If rounding is required, use the <code>Round</code> function. Example: <code>1 <> 4</code> Returns true <code>1 != 4</code> Returns true

Conditional Operator	Syntax	Logic
IsMissing	<p>IsMissing (<i>reference</i>)</p> <p>IsMiss (<i>reference</i>)</p>	<p>Tests if the reference contains a #missing result.</p> <p>Note: If the reference is an expanded row or column, then all resulting cells must be #missing in order for the condition to be true.</p> <p>Example:</p> <p>IsMissing([1])</p> <p>Returns true if row 1 has a #missing value.</p>
IsError	<p>IsError (<i>reference</i>)</p> <p>IsErr (<i>reference</i>)</p>	<p>Tests if the reference contains an #error result.</p> <p>Note: If the reference is an expanded row or column, all resulting cells must be #error in order for the condition to be true. Only formula rows and columns can result in #error.</p> <p>Example:</p> <p>IsError([2])</p> <p>Returns true if row 2 has a #error value.</p>
IsNonNumeric	<p>IsNN (<i>reference</i>)</p> <p>IsNonNumerid (<i>reference</i>)</p> <p>IfNN (<i>reference</i>)</p> <p>IfNonNumber (<i>reference</i>)</p>	<p>Tests if the reference contains a #missing or #error results.</p> <p>Note: If the reference is an expanded row or column, all resulting cells must be #missing and/or #error in order for the condition to be true.</p> <p>Example:</p> <p>IsNN([3])</p> <p>Returns true if row 3 has a #missing or #error value.</p>
Parenthesis	(<i>condition</i>)	<p>Groups a condition.</p> <p>Example:</p> <p>(1 > 4)</p> <p>Returns false</p>

Complex Conditions

Table 15 Complex Conditions

Complex Conditions	Syntax	Logic
And	<p>(<i>condition</i> AND <i>condition</i>)</p> <p>(<i>condition</i> & <i>condition</i>)</p>	<p>Compares two conditions. Returns true if all conditions are true.</p> <p>Example:</p> <p>(1 > 4 AND 5 > 2)</p> <p>Returns false</p>
Not	<p>NOT (<i>condition</i>)</p> <p>! (<i>condition</i>)</p>	<p>Negates the result by reversing the result of the condition.</p> <p>Example:</p> <p>Not (1 > 4)</p> <p>Returns true</p>

Complex Conditions	Syntax	Logic
Or	<pre>(condition OR condition) (condition condition)</pre>	<p>Compares two conditions. Returns true if any of the conditions are true.</p> <p>Example:</p> <pre>(1 > 4 OR 5 > 2) Returns true</pre>

Complex conditions And, Or, and Not are fully supported. However, they must be surrounded by parentheses.

Valid example:

```
If ( ([A] > [B] and [A] > 1000), [A], [B])
```

Invalid example:

```
If ( [A] > [B] and [A] > 1000, [A], [B])
```

Financial Functions

Subtopics

- [Rank](#)
- [Variance/Var](#)
- [VariancePercent/VarPercent](#)

Rank

Rank is a financial function that provides a rank value for a value in a specified range. Rank is processed by Oracle Hyperion Financial Reporting and does not depend on the database connection.

Syntax:

```
Rank([Reference], Order)
```

```
Rank([Reference], Order, Unique)
```

Argument	Description
<i>Reference</i>	<p>The range of cells, rows, or columns to rank, with letters identifying columns and numbers identifying rows. For example, specify [A,1:5] to rank the values for rows 1 through 5 in column A.</p> <p>You can use the .ifNN property with a range of cells to assign numbers to any cells with nonnumeric values so that those cells can be ranked. For example, you can use .ifNN(-1) to assign the value -1 to any cell with a missing value.</p>

Argument	Description
<i>Order</i>	<p>Indicates the order by which the values are ranked. The lowest value ranked in ascending order receives a rank result of 1. The largest value ranked in descending order receives a rank result of 1. The order can be indicated by any of the following keywords or values:</p> <ul style="list-style-type: none"> ● Ascending ● Descending ● Asc ● Des ● Desc ● 1 (the number 1 is the same as Ascending) ● 0 (zero is the same as Descending) <p>The keywords are not case-sensitive.</p> <p>Note: Do not enclose the number or keyword indicating order in quotation marks.</p>
<i>Unique</i>	<p>(Optional) A Boolean keyword indicating how to treat equal values in the Reference parameter where:</p> <ul style="list-style-type: none"> ● false (or omitted) – equal values receive the same ranking ; ranked results may be duplicated ● true – equal values receive a unique ranking; there are no duplicate rankings. Values in the Reference parameter are ranked on a first come, first ranked basis. For example, if values in rows 2 and 5 are equal, the value in row 2 is ranked before that of row 5.

Examples:

This formula in column B ranks the values in rows 1 through 5 in column A in descending order:

`Rank([A,1:5], descending)`

The result might be as follows:

	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	2
Diet	missing	missing
Root Beer	0	4

When two values are equal, they receive the same rank value. In the example above, Cola and Beer have the same value and therefore the same rank.

This formula in column B assigns the value of -1 to any nonnumeric value so it can be ranked:

`Rank([A,1:5].ifNN(-1), descending)`

In the following result, the missing value now has a rank of 5:

	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	2
Diet	missing	5
Root Beer	0	4

Example:

The following example builds on the previous example explaining how the new “unique” parameter affects the results:

This formula in column B assigns the value of -1 to any nonnumeric value so it can be ranked, and also indicates that each ranking should be unique:

```
Rank([A,1:5].ifNN(-1), descending, true)
```

In the following result, the missing value now has a rank of 5, and Beer has a value of 3 (even though it has the same data value as Cola):

	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	3
Diet	missing	5
Root Beer	0	4

Variance/Var

Subtopics

- [Expected Results Using Planning](#)
- [Variance Behavior](#)
- [Examples](#)

Variance is a financial function that evaluates the difference between the specified values based on account type for the current account. For example, for an Income, Flow, Asset, or Balance account, a positive result represents a favorable variance, so the result appears as a positive number. For Expense or Liability accounts, a positive result represents an unfavorable variance, so the result appears as a negative number. This function is available for standard Planning database connections.

Syntax:

`Var (reference1, reference2)`

where *reference1* and *reference2* are references to a row, column, or cell that correspond to members of the same Account dimension whose variance results are to be calculated.

Expected Results Using Planning

Table 16 Expected Results When Using *Variance* With Planning

Column A	Column B	Var ([A] , [B])=0	Var ([A] , [B])>0	Var ([A] , [B])<0
Asset	Asset	0	Returns a positive value	Returns a negative value
Liability	Liability	0	Returns a negative value	Returns a positive value
Equity	Equity	0	Returns a positive value	Returns a negative value
Revenue	Revenue	0	Returns a positive value	Returns a negative value
Expense	Expense	0	Returns a negative value	Returns a positive value

Variance Behavior

Variance expects comparison of the same account type. When you compare two different account types, like Sales & Expense, *Variance* performs the math without applying the logic of the account type. For example:

Sales	Expense	Result
-400	100	-500

Examples

Variance accepts cell, column, or row references only. For more information, see [“Row, Column, or Cell Reference Arguments” on page 87](#).

Syntax	Example
Sample syntax referencing a column:	<code>Var ([A] , [B])</code>
Sample syntax referencing a row:	<code>Var ([3] , [4])</code>
Sample syntax referencing a cell:	<code>Var (Cell [3,A] , [3,B])</code>

In this example, the variance between column A (**Actual**) and column B (**Budget**) is calculated as:

`Var ([A] , [B])`

This example produces the following report:

	Year	Product	Market
	Actual	Budget	Variance
	=====	=====	=====
Sales (Income)	400,855	373,080	27,775
COGS (Expense)	179,336	158,940	-20,396

VariancePercent/VarPercent

Subtopics

- [Expected Results Using Planning](#)
- [VariancePercent Behavior](#)
- [Examples](#)

`VariancePercent` is a financial function that evaluates the difference, in percent, between the specified values based on account type for the current account. For example, for an Income, Flow, Asset, or Balance account, a positive result represents a favorable variance, so the result appears as a positive number. For Expense or Liability accounts, a positive result represents an unfavorable variance, so the result appears as a negative number. This function is available for standard Oracle Hyperion Planning database connections.

Syntax:

`VarPer (reference1, reference2)`

where *reference1* and *reference2* are references to a row, column, or cell that correspond to members of the same Account dimension whose `VariancePercent` results are calculated.

Expected Results Using Planning

Table 17 Expected Results When Using `VariancePercent` with Planning Accounts Tagged With UDAs.

Col A	Col B	VarPer ([A] , [B])=0	VarPer ([A] , [B])>0	VaPer ([A] , [B])<0
Asset	Asset	0	Returns a positive value	Returns a negative value
Liability	Liability	0	Returns a negative value	Returns a positive value
Equity	Equity	0	Returns a positive value	Returns a negative value
Revenue	Revenue	0	Returns a positive value	Returns a negative value
Expense	Expense	0	Returns a negative value	Returns a positive value

VariancePercent Behavior

`VariancePercent` expects comparison of the same account type. When you compare two different account types, like Sales & Expense, the `VariancePercent` function performs the straight math without applying the logic of the account type. For example:

Sales	Expense	Result
-400	100	-5.

- #missing is treated as zero (0), unless specified differently using `ifnonnumber` property.
- #error is results in #error, unless specified differently using `ifnonnumber` property.

Examples

VariancePercent accepts, cell, column, or row references only. See [“Row, Column, or Cell Reference Arguments” on page 87.](#)

Syntax	Example
Sample syntax referencing a column:	<code>VarPer ([A], [B])</code>
Sample syntax referencing a row:	<code>VarPer ([3], [4])</code>
Sample syntax referencing a cell:	<code>VarPer (Cell [3,A], [3,B])</code>

In this example, the `VariancePercent` between column A (Actual) and column B (Budget) is calculated as follows:

`VarPer ([A], [B])`

This example produces the following report:

	Year	Product	Market
	Actual	Budget	VariancePercent
	=====	=====	=====
Sales (Income)	400,855	373,080	7%
COGS (Expense)	179,336	158,940	-13%