ADVANCED FEATURES FOR BUILDING CONTENT-RICH PORTALS WITH ORACLE9*i*AS

Jeanne Kramer, Oracle Corporation

INTRODUCTION

Effective management of content elements that make up a Web page has become increasingly important as the size and complexity of portal sites grow. In this paper, you will learn how to use the most powerful and flexible features of Oracle9iAS Portal Content Areas to manage Web content within your portal, including custom object types, extended attribution, external function integration, and other advanced object features.

Oracle Portal Content Areas provide a flexible framework for managing content driven web sites including a built in structure for organizing, classifying and cross referencing the web site's contents. In addition to providing the building blocks for a content driven web site which can be maintained via the web browser, Content Areas also provide ways to display the content using portlets in Oracle Portal Pages. Each of the Content Area objects which can be published as a portlet will be explored in this paper. A number of specialized portlets under the Content Area Provider such as Saved Searches and Owned Folders will also be addressed.

CONTENT CONTRIBUTION

ATTRIBUTES AND TYPES

Portal Content Areas give Administrators the control to designate the type of information that will be added to the portal. Each new element of content is added to a Content Area through the creation of an item. Nine basic types of items (such as 'File', 'Text', 'URL') and four basic types of folders are defined during installation.

Content Area items and folders must be based on an item or folder type. Each type has a specific assortment of attributes associated with it. For example, when a user adds an item of type *file* to a Content Area, several attributes specific to item type *file* will be presented for the user to define. These will include such attributes as the display name for the item, an item description, and an indication of when the item should expire.

Often a Content Area will need to track additional information at either the item or folder level. The installed default item types can be extended to create new item types with whatever additional attributes are needed. Custom attributes, item types, and folder types permit this type of customization. These new custom types may be created within a Content Area for use only in that Content Area or as a shared object which will be available for use in all Content Areas. Content Area Administrators may select the appropriate collection of item and folder types for use by a specific Content Area.

Through the creation of attributes and custom item and folder types, as well as the selection of those item and folder types to be used in a Content Area, Administrators have a great deal of control over what type of information is collected from content contributors.

Keep in mind that the values of your custom attribute can be added to the display of your items in folders. By modifying the style used for a folder and editing the region properties in the Folder Layout tab you can select and specify the display order for all attributes you want shown when items are displayed in that region.

Default Item Types

When adding content to a Oracle Portal Content Area folder, the user must select a distinct item type before entering the attributes of the content item. As mentioned above, each installed Oracle Portal provides an assortment of *default item types*. These include:

- *Application Component* provides a list of Oracle Portal application components, such as charts and reports. The selected application component is then exposed using the selected display option.
- *Image* provides the option to upload an image for display in the folder, either with or without image map information.
- *File* is used to permit the upload of a file/document.
- *Folder Link* is used to create a link to another Content Area folder.
- *Java Application* is used to host a Java Server pages application.
- *PL/SQL* provides the user with a text box for entering PL/SQL code. This PL/SQL code should generate HTML, which is displayed when the item is rendered.
- *Text* provides the user with a method of entering simple text into a Content Area item. This item type removes the need to create a separate file and upload it in order to add information.
- *URL* provides the user with a method of linking in external URLs into the folder.
- *Zip File* items are file items which provide the option to unzip the file within the product via an "Unzip" link beside the Zip File item. When the "Unzip" option is used the user may select a target folder, indicate if the unzip should occur in the background and select the file posting mode: "Overwrite Existing Files" and "Rename Extracted Files". Zip File items open many doors to the staging and maintenance of formerly static HTML web sites.

Once content contributors select an item type, they will be given a list of attributes to define for that item. It may be of interest to the Content Area Administrator to extend the item type definitions to add attributes and/or procedures. The following chart will help you understand those attributes that are assigned to each Content Area default item type. This table only shows attributes that are shared by more than one item type.

All the item types include additional special item type specific attributes. For example, the File item type includes an attribute for storing a file name and the Java Application item type includes attributes for storing the application name, jar file, initial page name and application type.

Attribute	Application Component	File	Folder Link	Image	Java App	PL/SQL	Text	URL	Zip File
Primary Item Attributes									
Folder Region	X	X	X	X	X	X	X	X	X
Display Name	X	X	X	X	X	X	X	X	X
Category	X	X	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X	X	X
Publish Date	X	X	X	X	X	X	X	X	X
Expiration Period	X	X	X	X	X	X	X	X	X

Attribute	Application Component	File	Folder Link	Image	Java App	PL/SQL	Text	URL	Zip File
Secondary Item Attributes									
Perspectives (multiple)	X	X	X	X	X	X	X	X	X
Image	X	X	X	X	X	X	X	X	X
Image Alignment	X	X	X	X	X	X	X	X	X
Rollover Image		X					X	X	X
Basic Search Keywords	X	X	X	X	X	X	X	X	X
Author	X	X	X	X	X	X	X	X	X
Enable Check Out	X	X	X	X	X	X	X	X	X
Hide Item	X	X	X	X	X	X	X	X	X
Display Options									
Item Displayed Directly In Folder Area	X					X	X	X	
Link That Displays Item In Folder Area	X					X	X	X	
Link That Displays Item In Full Browser Window	X					X	X	X	
Link That Displays Item In New Browser Window	X					X	X	X	
Display Parameter Form	X								

Figure 1: Shared attributes by base item type

The Content Area default item types are limited to the default attributes assigned to them. If you wish to append additional attributes to a default item type, you must create custom item types. Default item types cannot be altered. When you create a new item type based on a default item type, the default is copied, along with its related attributes. Then you can extend the copied item type by adding attributes that you create, but you cannot remove any of the default attributes. You can use extended item types to collect additional structured data via the item's attributes.

It is also possible to create item types that are not based on default item types. In this case, the item type will only begin with basic attributes, such Display Name, Category, and Perspectives. The item type can then be assigned additional custom attributes.

Default Folder Types

When adding a folder to a Content Area, the user must select a distinct folder type before entering the attributes of the folder. Each installed Content Area comes with an assortment of *default folder types*:

- *Container* is used to permit the creation and organization of content items.

- *PL/SQL* provides the user with a text box for entering PL/SQL code. This PL/SQL code should generate HTML, which is rendered when the folder is selected for viewing.
- *Search* permits creation of a folder based on a Content Area search. The search criteria is saved and executed each time the folder is rendered.
- *URL* provides the user with a method of linking in external URLs into the Content Area folder hierarchy.

All folder types have the same set of default attributes. It may be of interest to the Content Area Administrator to extend the folder type definitions to add attributes and/or procedures. Default attributes for folders include the following list.:

- Name
- Display Name
- Description
- Category
- Publish as Portlet
- Use Page Style (for Portlet)
- Portlet Display Name
- Item Versioning (folder level setting)
- Contact Email
- Enable Folder Caching
- Display sub-folders (permits selection of folders to display)
- Perspectives
- Style
- Title Image Name
- Rollover Image Name
- Banner Image Name
- Navigation Bar (navigation bar as separate entity)
- Access (item level security option)

PL/SQL, *URL* and *Search* folders are especially useful for extending your Content Area map to include additional information. *PL/SQL* folders give you complete control over the display of database stored information. *URL* folders give easy integration with and migration from existing Web sites. *Search* folders enable you to create different views of your Content Area for different audiences. For example, each category and perspective can be surfaced on the Content Area map as a simple *Search* folder.

ATTRIBUTES

To extend the default item and folder types provided by Content Areas, you must create new attributes. Attributes are custom fields that the Content Area administrator creates to capture more data about an item or folder. These attributes can be displayed when an item or folder is displayed. These attributes can also be passed to external procedures to integrate your Web site with other applications. You can reuse attributes that you create across multiple item and folder types. This means that if you add a Due Date attribute and use it in two different item types, a search for items with a specific due date will examine all items that have a due date attribute.

You can add the following types of attributes:

- Boolean
- Date
- File
- Number
- PL/SQL
- Text
- URL
- Application Component

For each attribute, the following properties are available:

- *Datatype* means one of the attribute types listed above.
- *Name* is an internal name, and must not have spaces or special characters.
- *Display Name* is the displayed attribute label.
- *Length* describes an item's length, when appropriate.
- *Enable Translations* answers the question: If the object which has this attribute is translated, should the value of this attribute be translated? If this option is checked, then the attribute will be permitted to have a different value for each translation. If this option is not checked, then the attribute's value will remain the same across all translations. The default for this option is checked.
- *Display Options* are appropriate depending on the attribute type (such as Single-Line, Multi-Line, List of Values).

Attribute List of Values

Each custom attribute you create for use with extended item and folder types can use a list of values (LOV). The LOVs used for custom attributes must first be created in the Application Building portion of Oracle Portal.

The following are the steps required to create an LOV based on a static list of values for use by a Content Area Attribute:

- First decide if you would like your LOV application to be created within the portal schema or in a new schema. In the case in which you would like to keep your Application and LOVs in a new schema, you must create a new schema (note that this is different from creating a new user and is performed from a link located under the 'Administer Database' tab of the design time home page) for this purpose
- Create a new application. Select either your portal schema or the new schema you created in the last step to own the application.
- After creating the application, use the Navigator to drill into the application in order to add components to it.
- Select LOV from the links beside the 'Create New…' prompt near the top of the screen
- Select 'Static List of Values' to create a new list of values
- Enter a name, values for your LOV and designate your LOV properties. Save your LOV.
- Now it is time to create an attribute. Decide if your attribute should be shared across content areas, or only used in a single content area. Once you have decided, drill into the 'Custom Types' portion of the content

area (this can be done either in the Navigator or using the Administration page for the Content Area). Also make sure that the type of attribute you create (text, number, date… etc) matches the data type of the value returned by your LOV.

- The new attribute can now be added to a custom item or folder type on the Attributes tab in the manager for an existing custom item or folder type.
- Finally, to see how your LOV is working - use your new custom type to create an object in the content area

Extended Item and Folder Types

An extended item type must be based on one of the default item types discussed above in the "Default Item Types" section. An extended folder type must be based on one of the four default folder types discussed above in the "Default Folder Types" section. For each item and folder type added, the following properties are available:

- *Name* is an internal name, and must not have spaces or special characters.
- *Display Name* is the displayed label for the type.
- *Description:* text description of the type.
- *Icon:* an image for display that represents the type.
- *Selected Attributes:* the attributes which have been added to the item type being extended. Each attribute added to the item type may be assigned a default value and marked to be displayed either on the Primary or Secondary tab in the object manager.
- *Procedures:* external function calls. Each procedure may pass a different collection of attributes with procedure specific parameter names. Reference the "Procedures" section below for more details.

Let's look at an example of an extended item type. Perhaps a Web site has been created at a company that publishes magazines. It might be useful for this Web site to have a type of item based on the file item type, called "Magazine Article." In addition to all of the attributes that are included with the *file* default item type, the Content Area administrator may wish to define custom attributes to capture structured data about a "Magazine Article." These might include editor, proof reader, and due date. For every new "Magazine Article" item, Oracle9*i*AS Portal will prompt the user for all the required attribute data, including the editor, proof reader, and due date for the magazine article.

The display of an attribute is determined by the display rules that apply to the region in which the attribute will appear, which in turn is dependent on the style selected for the folder. Refer to the "Folder Display Rules and Styles" section below for more details.

Extended Item and Folder Type Procedures

Another aspect of custom item types is the ability to associate external functions (be they HTTP or PL/SQL functions) with a custom item type. These functions can pass values of attributes at the item level via a link which can be displayed along with other item attributes by modifying the Folder Layout Region Properties of the style used for a folder.

Defining a procedure for a type means defining a call to an *external* function. It should not be confused with creating an item of type "PL/SQL call" or "Application Component." An item type or folder type procedure permits you to create a connection between every specific instance of an item or folder type and an existing procedure, which may be

called via PL/SQL or HTTP. You can define a procedure that will select and pass attributes to an external procedure at runtime. Each item or folder type can have multiple procedures associated with them.

When defining a procedure, the Content Area administrator may use the following attributes to affect the way the procedure behaves.

- *Type*: HTTP or PL/SQL. This determines the syntax of the procedure call.

  An example of an HTTP procedure call:

  ```
  http://www.graphics.com/makegraph.html
  ```

  An example of a PL/SQL procedure call:

  ```
  scott.paintchart.salesreport
  ```

- *Procedure Call*: The actual code for the function call itself. The syntax of the function must match the selection of the function type above.
- *Link Text*: This is the text of the link that will be displayed beside the instance of the item or folder type. This link text is displayed only if you do not select Display Procedure Results With Item. The user can click on this link to execute the procedure.
- *Icon*: An image file. If specified, this image will be displayed in place of the link text.
- *Display Procedure Results With Item*: Indicates if the procedure should be automatically executed and the results displayed with each instance of the item or folder type.
- *Attributes*: Content Areas permits you to pass any attributes of the type to the procedure. It also permits the specification of a different parameter name for each attribute in each procedure.

Let's look at an example of an extended item type which has a procedure. In our example, a custom item type of "Sales Report" has been created. This custom item type is based on the *file* item type. Two attributes have been added to this item type: "Sales Manager" and "Period Ending Date". They are both required. There is another web-based system which will generate a graph displaying the current industry breakdown of a sales report given the name of the sales manager and the period ending date. The Content Area Administrator has created a function for this item type which is a call to the graph. Both the Sales Manager and Period Ending Date attributes have been selected to be passed to this function. The function has "Display Procedure Results With Item" selected.

Each time a user adds a new Sales Report to a folder, they will be prompted to enter all the required values for creating a file type item as well as values for Sales Manager and Period Ending Date. Whenever an item of this type is displayed the function will be executed using the Sales Manager and Period Ending Date for that specific item. The results of the function call, in this case the graph of the current industry breakdown for the sales report, will be displayed on the web page below the displayed attributes for the Sales Report.

FOLDER DISPLAY RULES & STYLES

Oracle Portal Content Areas provide a great deal of flexibility to the layout and style properties of Folders. A style is used to designate the colors, fonts, and display rules for folders and navigation bars. By modifying the *Properties* and *Folder Layout* of a style used by a folder, you can dramatically change the display of a folder.

The *Region Properties* portion of the style governs the display of the items on the page. There are five folder layout options (Figure 1). These layouts divide the content area of the folder display into five regions.

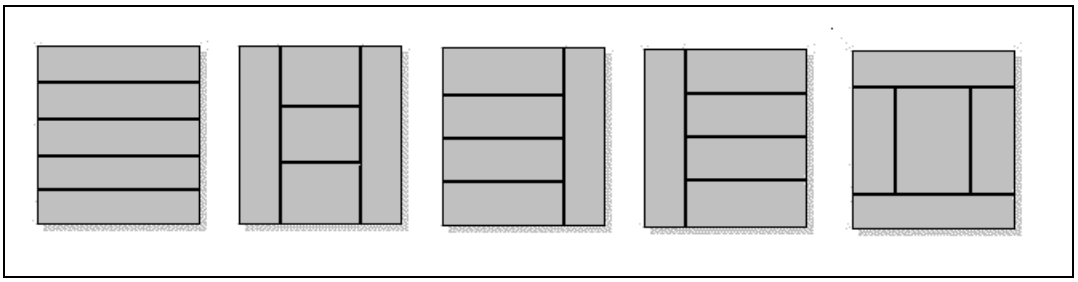

Figure 1: Available Folder Layouts

You can set the following properties for each of the folder regions:

- Folder Region Display Name: Users can refer to the region name to determine where to add items.
- Folder Region Type: A folder region may display either Items or Sub-Folders. There may be only one Sub-Folder region per style.
- Display Region Banner: This indicates if a banner should be displayed above the region displaying the region's display name.
- Displayed Attributes: This is where you indicate which attributes should be displayed for each item assigned to the region. What you specify for this value includes display order of selected attributes.
- Group By: Group displayed items by a selected attribute. For example, the items in a region may be grouped by category, author, date or item type.
- Display Group By Banner: If the items in a region are grouped by an attribute, this indicates if a banner should be displayed above each group of items. If so, there will be a link on the banner to permit viewing of all items possessing the selected attribute across the site.
- Sort By: Sort items by a selected attribute.
- Sort Order: Arrange items in ascending or descending order.
- Item Icon Height (pixels): If entered, will force all item icons displayed in the region to be of the designated height. If left blank, item icons displayed in the region will be displayed with their original height.
- Item Icon Width (pixels): If entered, will force all item icons displayed in the region to be of the designated width. If left blank, item icons displayed in the region will be displayed with their original width.
- Alignment: indicates alignment (such as left, right, center, etc…) of items displayed in region.
- No. of Columns: The number of columns of items or sub-folders. Will not limit the number of items displayed.
- No. of Rows: The number of rows of items or sub-folders. If a value is entered for "No. of Rows", it will limit the number of items displayed (columns x rows).
- Background Color: Background color to display for the region.

Remember that the folder layout and region properties of the style only govern the display of items within a folder. Modifications to the folder layout and region properties will not affect a navigation bar using the style. Content Areas also permits the movement of items between regions.

By applying folder layouts and region properties in a variety of combinations, the final look of any folder can be extensively customized.

SEARCHING YOUR CONTENT

Oracle Portal provides a number of ways to customize the search experience for your users. You can use portlets from which users can submit search requests as well as modify the look of the page on which searches are displayed. All the search portlets described below are

SEARCH PORTLET

This portlet permits users to launch a Content Area search directly from a portlet. The portlet permits the end user to select the content area within which to search and provides a text box for the entry of search terms. Links are also provided for accessing search tips and the advanced content area search screen. No customization is available for this portlet. The results of searches launched from this portlet will be displayed on the Search Page described below.

SEARCH RESULTS PAGE PORTLET

This new portlet, introduced in Oracle Portal 3.0.9, permits some customization of the way that search results are displayed. This portlet is used on the Search Page which is provided out of the box during installation (see next section for more details about the Search Page).

SEARCH PAGE

You can customize the page on which search results are displayed. To send the search criteria from all basic search boxes on navigation bars, as well as all advanced search screens to a new page - change the value for the 'Specify Search Results Page' setting in the Search Settings (which has a link in the Services portlet on the Administration tab of the design time home page).

SAVED SEARCHES PORTLET

This portlet gives users access to their saved searches. Each user will see something different in this portlet - it will show them their own saved Content Area searches. Clicking on the link for a saved search in the portlet will run the search again and retrieve current search results.

Customizations available for the Saved Searches portlet:

- Display Name
- Order Saved Searches by Creation Date or Name (Ascending or Descending)
- Display all Saved Searches or only a specific number

Add a new search directly from the customize Saved Searches portlet screen

USING THE NAVIGATION BAR PORTLET

You can take advantage of Navigation Bar portlets to create a portlet that just provides access to one of the 'special' elements that can be added to Navigation Bars. For example, a Basic Search box and Advanced Search link (both 'Basic Elements' that can be added to a Navigation Bar) could be added to a new Navigation Bar. This Navigation Bar can then be published as a portlet and placed on any page from which you want to provide easy access to Oracle Portal search.

ADVANCED SEARCH API

Currently the Advanced Search screen is not customizable via the user interface (Figure 1), but there is a method for customizing the look of this screen to suit your own requirements. Using a combination of a published public API and a search setting at the portal level, you can create your own look and feel for this page. Search results are returned on your selected Search Page as described in the section above. You cannot add functionality that is not on this screen

(such as an option to select more than one Perspective to search on), but you can make a screen that arranges these search criteria (or any subset of them) in any way you like.

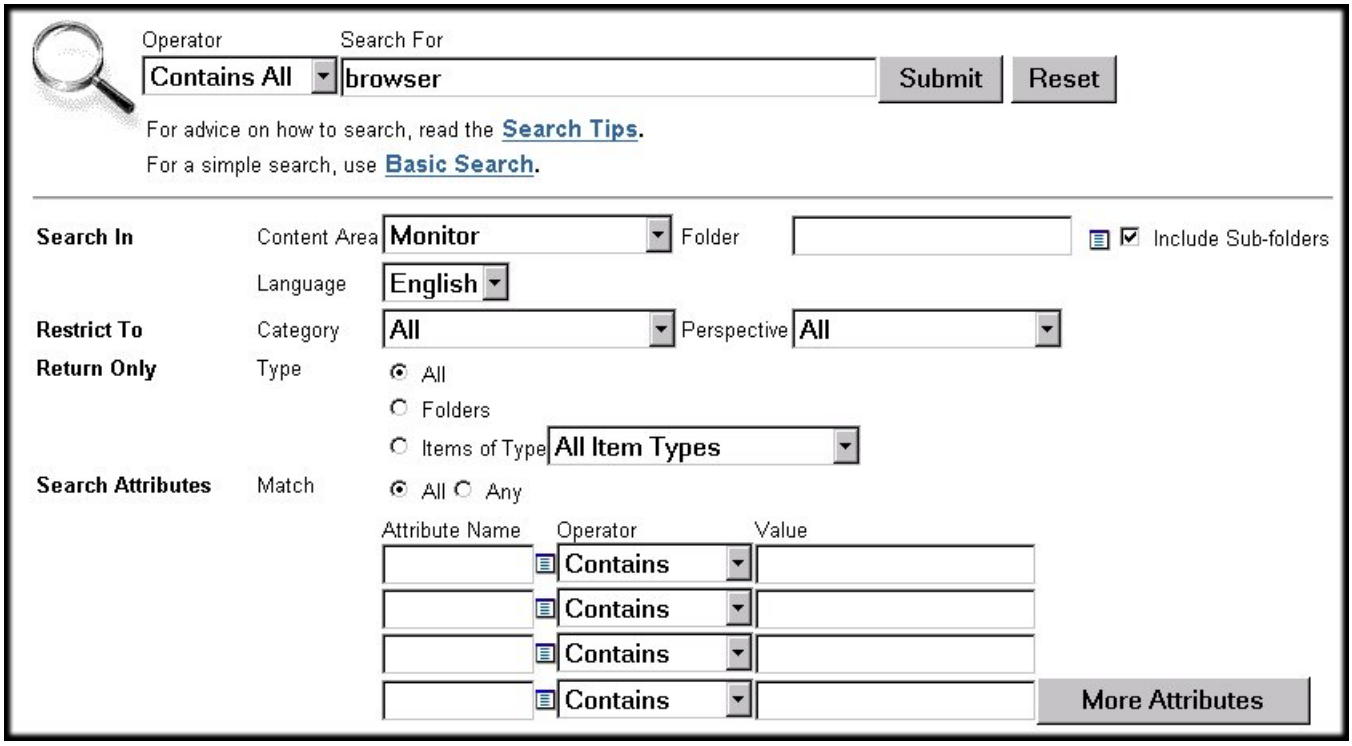

Figure 2: Advanced Search Screen in Oracle Portal

Once you have created your new user interface which passes the search criteria to the API explained below, you must add the URL for your new Advanced Search code to the Search Settings (which has a link in the Services portlet on the Administration tab of the design time home page). This URL goes in the Advanced Search Link setting.

Another way to use the API outlined below is to write a portlet which shows some assortment of search criteria fields. This portlet could then call the submit search procedure below and the search results would then be shown on the same Search Results page described in the section above.

The API which has been created to permit you to submit search criteria to the portal as though it was submitted from the current hard-coded Advanced Search page has the following specification:

```
Package Name: wwsbr_search_api

Constants:
  YES constant number := 1;
  NO  constant number := 0;

  -- search operator constants
  FIND_ANY constant varchar2(10) := 'ANY';
  FIND_ALL constant varchar2(10) := 'ALL';

  -- search for type constants
  ALL_OBJECTS     constant varchar2(10) := 'ALL';
  FOLDERS         constant varchar2(10) := 'FOLDERS';
  ITEMS           constant varchar2(10) := 'ITEMS';
```

```
  -- attribute operator constants
  CONTAINS_ALL    constant varchar2(20) := 'containsall';
  GREATER_THAN    constant varchar2(20) := 'greaterthan';
  LESS_THAN       constant varchar2(20) := 'lessthan';
  EQUAL_TO        constant varchar2(20) := 'equal';

procedure submit_search (
    p_search_terms in varchar2 default null,
    p_search_operator in varchar2 default FIND_ANY,
    p_caid         in number default 0,
    p_current_caid in number default null,
    p_language     in varchar2 default null,
    p_folder_id    in number default null,
    p_folder_caid    in number default null,
    p_item_type_id in number default null,
    p_include_child_folders in number default NO,
    p_category_id  in number default null,
    p_perspective_id in number default null,
    p_search_for_type in varchar2 default ALL_OBJECTS,
    p_attribute_id wwsbr_type.array default wwsbr_type.empty,
    p_attribute_name wwsbr_type.array default wwsbr_type.empty,
    p_attribute_caid wwsbr_type.array default wwsbr_type.empty,
    p_attribute_data_type wwsbr_type.array default wwsbr_type.empty,
    p_attribute_operator wwsbr_type.array default wwsbr_type.empty,
    p_attribute_value wwsbr_type.array default wwsbr_type.empty,
    p_style_id in number default null,
    p_style_caid in number default null
   );
```

Parameter	**Description**	**Valid Values (all caps terms are constants)**
p_search_terms	terms to search for	Default: Null
p_search_operator	search operators.	Default: FIND_ANY FIND_ANY or FIND_ALL
p_caid	content areas to search within	Default: 0 0 = all content areas any other number = content area's id
p_current_caid	number the content area id for the content area to be linked to under the content area home link in the search results screen	Default: Null
p_language	language to search	Default: Null
p_folder_id	search in the folder with this id	Default: Null
p_folder_caid	content area id for p_folder_id	Default: Null
p_item_type_id	return only items of this item type (assumes content area id of p_caid)	Default: Null
p_include_child_folders	if searching in a specific folder, should the child folders be included in the search	Default: NO
p_category_id	restrict search to objects with this category. Assumes content area id of p_caid.	Default: Null
p_perspective_id	restrict search to objects with this perspective. Assumes content area id of p_caid	Default: Null

Parameter	Description	Valid Values (all caps terms are constants)
p_search_for_type	Search for items & folders or items or folders	ALL_OBJECTS or ITEMS or FOLDERS
p_attribute_id	wwsbr_type.array array of attribute ids for to search for specific attributes with specific values	Default: wwsbr_type.empty
p_attribute_name	wwsbr_type.array array of attribute names	Default: wwsbr_type.empty
p_attribute_caid	wwsbr_type.array array of attribute caids for item based on custom type which has attributes.	Default: wwsbr_type.empty These values must either match p_caid or be 0
p_attribute_data_type	wwsbr_type.array array of attribute type	Default: wwsbr_type.empty values ('url','text'). this value is in WWSBR_ATTRIBUTES.DATA_TYPE
p_attribute_operator	wwsbr_type.array array of search operators	Default: wwsbr_type.empty CONTAINS_ALL GREATER_THAN LESS_THAN EQUAL_TO
p_attribute_value	wwsbr_type.array array of attribute values to search for in the corresponding attribute	Default: wwsbr_type.empty
p_style_id	id of style to use for rendering search results	Default: Null
p_style_caid	content area id of style to use for rendering search results	Default: Null

PUBLISHING YOUR CONTENT

As explained above, in Content Areas items are added to a *Folder*. These folders are very much like folders in a file system, with multiple items existing within a folder Folder and item access may be explicitly granted to users or groups of users. Both items and folders are classified by assigning them a *Category* and one or more *Perspectives*. *Navigation Bars* are collections of navigation elements which can include links to folders, categories, perspectives and anything else you wish.

Folders, Categories, Perspectives and Navigation Bars may all be published as portlets. This option permits those users with appropriate portlet publication privileges to make information stored in a content area available to the entire Oracle Portal community. In this way, content such as lists of documents, status reports, application components and organized URL links can be published for use on multiple pages in the portal.

This section will explore the basic ways to using Content Area Portlets as well as creative ways to take advantage of them.

Content Area Portlets: General Concepts

Publishing vs Selecting

There are two ways to place Content Area objects on a page as a portlet – you may either publish the object as a portlet OR customize a generic object portlet to show the specific content area object you desire to display.

Publish an Object as a Portlet

Folders, Categories, Perspectives and Navigation Bars can all be published as portlets. This is done in the Object Manager for the chosen object. For Folders and Navigation Bars, the user must have Own privileges on the object or Administration privileges on the Content Area or Portal. For Categories and Perspectives, the user must have Administration privileges on the Content Area or Portal. When a portlet of this type is added to a page, the page creator cannot change the specific object that is displayed; for example, the page creator cannot customize a category-specifc portlet to use a different category.

Generic Object Portlets

There are four special Content Area portlets which permit users to select the specific object to publish when the portlet is added to the page, one for each of the different objects one can display in a portlet: Folder, Category, Perspective and Navigation Bar. These generic portlets permit selection of which specific object is to be displayed in the portlet via editing the portlets defaults or customizing the portlet.

Styles (Colors and Fonts)

All of the portlets displaying Folders, Categories, Perspectives and Navigation Bars use their corresponding objects' setting for 'Use Page Style' which is set in the object manager for each of these objects. If this option is checked in the object manager then the style of the page on which the portlet is displayed is inherited by the portlet. If this option is not checked, then the style assigned to the object in the Content Area will dictate the colors and fonts whenever the portlet is displayed regardless of the page style.

For example, let us assume a Folder named 'Entertainment' exists. If this Folder was published as a portlet and the 'Use Page Style' option was checked, then every time this folder portlet is put on a page, it will inherit the colors and fonts from the page style for the page on which it is displayed. Each instance of this folder portlet on different pages with different styles will be rendered differently.

In another example, let us assume a Folder named 'Map' exists. If this Folder was published as a portlet and the 'Use Page Style' option was NOT checked, then every time this Folder portlet is put on a page it would always use the colors and fonts of the style assigned to the 'Map' Folder in the Content Area.

Privileges Required

In order to publish any Content Area objects as portlets, a user must have the Publish Portlet privileges.

In order to select an object for display in a generic object portlet, the user must have appropriate privileges to view the object.

Content Area Providers & Portlet Repository

After at least one object from a Content Area has been published as a portlet, that Content Area becomes a Portlet Provider. Each Content Area provider will be added as a folder in the Portlet Repository.

There is also a basic Content Area provider which provides the generic object portlets, as well as a number of special purpose portlets.

The Portlet Repository is a Content Area which can be modified to change the portal user's experience for finding portlets. Keep in mind that given the proper privileges, you can reorganize the Portlet Repository at any time. The folder created for Content Areas can be moved, privileges for viewing them can be modified and portlet items can be relocated into other folders entirely. Because it is a Content Area, the Portlet Repository gives appropriately privileged users the power to modify what is viewed by users in the portal.

Content Area Page

Each Content Area has a special portal page associated with it. This page's layout and portlets are reused for displaying all folders in the content area. The content of the folder will be displayed wherever the folder portlet is placed on the page. The default layout of this page is to have a Navigation Bar portlet on the left side and a Folder portlet on the right. The Navigation Bar portlet will display the navigation bar associated with the Folder (as designated on the Navigation Bar tab of the Folder manager). To edit the Content Area's page, click on the 'Page' tab of the content area manager, which can be reached either from the Administration menu for the content area or by clicking on 'Edit Properties' for the content area in the Navigator.

The Navigation Bar portlet that is displayed next to folders by default on the Content Area page can be removed by actually deleting the Navigation Bar portlet on the Content Area page. This can help when trying to make your content area page look and feel the same as the rest of your portal pages. It is also possible to do the reverse - add a Navigation Bar portlet to your other portal pages to provide a unifying navigation model across both your content areas and other portal pages. See the details about the Navigation Bar portlet below for more information.

Folder Portlets

Security

The security of a Folder is maintained in the folder portlet. If a user does not have the privileges to view a folder then the following rules apply:

- That user cannot add that folder as a portlet on a page, even if the folder has been published as a portlet.
- That user will not be able to select that folder for display in a generic folder portlet.
- If the user views a page that contains a portlet displaying that folder, the user will not see the portlet at all.

Folder Portlet Customization

All folder portlets have at least two kinds of customization available.

- First, the display name of the portlet can be modified.
- Second, the regions of the folder which are displayed in the portlet can be selected.

Tips for Folder Portlet Use

There are many ways to take advantage of folder portlets. They can be used for their traditional content management features, but they can also be used as quick and easy ways to get specific content onto your portal pages without writing portlets or custom code.

Let's say you want to add a quick list of URL links to your page. Here are the steps you could follow to use a folder portlet to do this:

- First, create a folder. If you have a personal folder, you can create your new folder inside it. If you don't have a personal folder, you will need to get privileges to create a folder within an existing content area.
- Second, add URL items to that folder
- Third, determine how you want the items displayed. To change the attributes displayed in the folder for these new items edit the folder style and click on the Folder Layout tab of the style. Now click on the edit icon for the region in which your URL items are currently residing. This will give you a pop-up window in which to designate the attributes (such as display name, description, etc) to be displayed. You can also change the number of columns or rows your items are displayed in.
- Fourth, when you have your items displaying properly, edit the folder's properties and publish the folder as a portlet. (Remember that you need special privileges to publish objects as portlets.)
- Fifth, add the folder portlet to your page. To find your new folder portlet in the portlet repository window used for adding portlets, look under the 'Other Providers' grouping of folders. Click on the name of the

content area in which your folder was created. If you created your folder in your personal folder, check in the 'Portal Content Area' provider.

- Finally, customize (for others) the folder portlet by editing the folder portlet's defaults. You can change the portlet's title as well as specifying which folder regions (these correspond to the folder layout regions in the style as well as corresponding to the regions you select from when adding content to your folder) to display.

This technique can also be very useful for adding images and text to a portal page. Combined with modifying the page's region properties (such as if the portlet's header and borders should be displayed), this technique for using folder portlets can help in building any portal page you can imagine while not requiring you to learn how to build portlets from scratch.

Category & Perspective Portlets

Categories and Perspectives are used in Oracle Portal to classify items and folders. These classified items can be viewed together using Category and Perspective portlets. For example, if a perspective exists for 'Outdoor Entertainment', then a portlet for that Perspective can be published so that page builders can put a list of all Outdoor Entertainment items on any page.

Security

Categories and perspectives do not have any security associated with them, but the items assigned to a category or perspective have security. The items displayed in a category or perspective portlet retain the security they have within the Content Area.

Portlet Customization

Category and Perspective portlets permit users to customize the number of items displayed within the portlet on the page as well as the display name for the portlet.

Navigation Bar Portlets

Basic Concepts

A Navigation Bar is made up of a collection of Navigation Bar Elements. Examples of such elements include links to Folders, images, links to any URL, the Content Area Logo, a simple search box or simply a line or two of text.

Security

If a Navigation Bar has been made Public, then any user may select it for display in a Generic Navigation Bar portlet. If a Navigation bar is not Public, then only the user who created it may select it for display in a Generic Navigation Bar portlet. Keep in mind that the security of a Navigation Bar only controls who can actually choose the Navigation Bar for display in a Generic Navigation Bar portlet. Once the Navigation Bar has been selected for display, any user with appropriate privileges to view the page which holds that portlet will be able to view the Navigation Bar.

Navigation Bar Portlet Customization

The only element that can be customized for a Navigation Bar portlet is the title displayed on its banner.

Tips for Navigation Bar portlet use

Often Navigation Bars are used to help users find their way through a Content Area, but it is completely reasonable to create a Navigation Bar purely for aiding in navigation of elements not related to a Content Area. Because you can add images, URLs and text to a Navigation Bar, you can use them to create navigation objects which may not even have any references to Content Area objects.

General Tips

Customize for Myself vs Customize for Others

When customizing a portal page, you have the option of customizing the page (and the portlets on the page) for 'Myself' and for 'Others'. Customizing the page for others means that you are changing the basic definition of the page so that anyone who views the page will see the changes. Customizing the page for yourself means that you are only modifying the way that the page will be displayed for you. Other users will still see the basic definition of the page until they choose to customize the page for themselves.

Direct Access URLs

Direct access URLs provide an easy way to create links directly to portal objects. They can be used in Folder URL items and Navigation Bar URL elements to provide links to Folders, Categories, Perspectives and Pages. These URLs can then be published via either a folder or navigation bar portlet to provide simple access to a collection of objects on a portal page.

Basically direct access URLs are built as follows:

```
http: //<hostname:port>/pls/<DAD>/url/<object type>/<path>/<object name>/
```

In this example, the string 'url' is a keyword that indicates that what follows is a direct access path. The keyword can be customized to be something other than 'url' if that is preferred. This would be done in the DAD customization screens.

The string 'folder' indicates that the object this URL points to is a folder.

The string 'entertainment/music/rock' represents the tree of internal reference names that point to the rock folder which is in the music folder in the entertainment content area.

The direct access url can be found in the property sheet of Folders, Categories and Perspectives. For pages, use the path shown in the page manager to build your URL (using the 'page' keyword in place of the 'folder' keyword in the example above).

- <hostname:port> is the name and HTTP port of your portal server
- pls tells the apache listener that this is a mod_plsql url
- <DAD> is the Database Access Descriptor that is used to locate the portal on your server; the default DAD is "portal30"
- url is a literal keyword, indicating that what follows is a direct access path. The keyword can be customized to something other than 'url', if that is preferred. This would be done in the DAD customization screens.
- <object type> is the type of object referenced in the URL. Allowable types are "page", "folder", "category", and "perspective"
- <object path> is the path of names that uniquely identifies this object. Note that this path uses the internal, unique names, not the display names.
- <object name> is the name of the object. Note that this is the internal, unique name, not the display name.

For example, to access the subfolder "rock", within the folder "music", in the content area "entertainment", on a server called www.mycompany.com that listens on port 8080 and has a DAD of "hqportal", the direct access URL will be:

```
http://www.mycompany.com:8080/pls/hqportal/url/folder/entertainment/music/rock/
```

CONCLUSION

By using the advanced tools that Content Areas provide, including classification, extensible attribution, search, and style management, Administrators and Portal page builders can leverage the power of Oracle9*i*AS Portal to create a custom web based content management and publishing environment.