CHAPTER 1

Out-of-the-Box Oracle Application Server Portal Technologies
Oracle defines Oracle Application Server Portal as “a rich, declarative environment for creating a portal Web interface, publishing and managing information, accessing dynamic data, and customizing the portal experience, with an extensible framework for J2EE-based application access.” While that one-sentence summarization is quite a mouthful, it does not do OracleAS Portal justice. OracleAS Portal is an incredibly powerful environment that enables developers to create and test sophisticated applications, all while writing and maintaining a very small amount of code, if they desire. Like any truly powerful development environment, OracleAS Portal can be used in a multitude of ways. For example, beginning developers can use OracleAS Portal’s wizards to generate applications consisting of forms, reports, and graphs and deploy those Oracle Application Server components quickly, easily, and with a minimum of code. Advanced developers can enhance the generated components through the use of the OracleAS Portal Application Programming Interface (API) or even bypass the wizards altogether and use Java and/or PL/SQL to create OracleAS Portal portlets. The portal you create can pull data from a single database, from multiple databases (including non-Oracle databases), and even from other sites on the Web. OracleAS Portal can be used to create portlets that interact with other sites on the Web and use their content in your portal. This chapter will briefly discuss all of the major components of OracleAS Portal. Subsequent chapters will go into the major topic areas in greater detail. This chapter also outlines how to create portlets with the wizards built into OracleAS Portal. The development chapters that form most of this book, however, will focus on using OmniPortlet, Oracle JDeveloper, and the OracleAS Portal Development Kit (PDK) to create advanced portlets.

OracleAS Portal leverages open standards, enabling developers to build Java 2 Platform, Enterprise Edition/eXtensible Markup Language (J2EE/XML) components that can be exposed within the framework as pure HTML. With the introduction of Web Services for Remote Portals (WSRP) and Java Specification Request (JSR) 168, OracleAS Portal support includes the capability to build interoperable applications that can be deployed across multiple vendor platforms. Furthermore, since OracleAS Portal is a component of the Oracle Application Server, it can integrate with other components such as Oracle Application Server Discoverer and Oracle Application Server Reports to expose rich Business Intelligence Reports. Chapter 11 discusses integrating Oracle’s other development tools (Oracle Forms, Oracle Reports, and Oracle Discoverer) into OracleAS Portal. As part of the Oracle Application Server, OracleAS Portal can also be deployed in a number of different architectures to support scalability and high-availability scenarios. Chapter 9 will discuss some of the options available to OracleAS Portal administrators.
Logging in to OracleAS Portal for the First Time

When Oracle Application Server 10g is installed with either the Portal and Wireless or Business Intelligence and Forms option, OracleAS Portal is installed also. Before we can access the OracleAS Portal environment, the necessary components must be up and running. Use the opmnctl tool (in the $ORACLE_HOME/opmn/bin directory) to see if the OracleAS Portal Oracle Application Server Containers for J2EE (OC4J) container is running:

```
O:\MT_HOME\opmn\bin>opmnctl status
```

```
Processes in Instance: MT_HOME.oski-2k3
-------------------------------+--------------------+---------+---------
ias-component              | process-type       |     pid | status
-------------------------------+--------------------+---------+---------
DSA                        | DSA                |     N/A | Down    
LogLoader                  | logloaderd         |     N/A | Down    
dcm-daemon                 | dcm-daemon         |    3584 | Alive   
OC4J                       | home               |    5940 | Alive   
OC4J                       | OC4J_Portal        |    2256 | Alive   
OC4J                       | OC4J_BI_Forms      |    3664 | Alive   
WebCache                   | WebCache           |    2756 | Alive   
WebCache                   | WebCacheAdmin      |    2616 | Alive   
HTTP_Server                | HTTP_Server        |    2820 | Alive   
Discoverer                 | ServicesStatus     |    2884 | Alive   
Discoverer                 | PreferenceServer   |    2904 | Alive   
wireless                   | performance_server |       0 | NONE    
wireless                   | messaging_server   |       0 | NONE    
wireless                   | OC4J_Wireless      |    4116 | Alive   
```  

If you are running on a server that has both the infrastructure and the middle tier on the same machine, make sure you run this command from the ORACLE_HOME location of the middle tier. If the OC4J_Portal component is not running for any reason, you can start it by executing the following command:

```
opmnctl startproc ias-component=OC4J
```

OracleAS Portal also depends on the security functions built into Oracle Application Server 10g. All of the security features of Oracle Application Server 10g are contained in the infrastructure. The infrastructure is another instance of the Oracle Application Server installation that can either reside on the same server as the mid-tier (where Portal lives) or on another server. To check to make sure that the
security piece is up and running, cd to the ORACLE_HOME of your infrastructure, then to the opmn\bin directory. Execute the following command:

```
opmnctl status
```

<table>
<thead>
<tr>
<th>ias-component</th>
<th>process-type</th>
<th>pid</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSA</td>
<td>DSA</td>
<td>N/A</td>
<td>Down</td>
</tr>
<tr>
<td>LogLoader</td>
<td>logloaderd</td>
<td>N/A</td>
<td>Down</td>
</tr>
<tr>
<td>dcm-daemon</td>
<td>dcm-daemon</td>
<td>5644</td>
<td>Alive</td>
</tr>
<tr>
<td>OC4J</td>
<td>OC4J_SECURITY</td>
<td>4056</td>
<td>Alive</td>
</tr>
<tr>
<td>HTTP_Server</td>
<td>HTTP_Server</td>
<td>1104</td>
<td>Alive</td>
</tr>
<tr>
<td>OID</td>
<td>OID</td>
<td>2596</td>
<td>Alive</td>
</tr>
</tbody>
</table>

If the OC4J_SECURITY component is not running for any reason, start it with the following command:

```
opmnctl startproc ias-component=OC4J
```

To access OracleAS Portal for the first time, enter the URL in your web browser from the following template:

```
http://<server>:<port>/pls/portal
```

The machine I used for the examples in this book is called oski-2k3, and the middle tier was installed on port 80 (since 80 is the default for http, you do not have to specify it in your URL), so to access OracleAS Portal on this server:

```
http://oski-2k3/pls/portal
```

During installation, Oracle will attempt to use port 7777 for your Oracle Application Server 10g infrastructure instance. If, for whatever reason, port 7777 is not available, Oracle will try 7778, then 7779, and so on. If you chose to install both the infrastructure and middle tier on the same machine, then, most likely, the infrastructure will use 7777 and the middle tier will use 80. OracleAS Portal “lives” in the middle tier, so use the middle-tier port number to access it. It is important to note that while Oracle Application Server’s Portal Page Engine (PPE) lives in the middle tier, it is dependent on both the middle tier and the infrastructure for both metadata and identity management.

If all of the necessary components are up and running, you should see a page similar to the one shown in Figure 1-1.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

Getting Around in OracleAS Portal

As with virtually every OracleAS Portal page developers or administrators will work with, a lot of information is displayed. The page in Figure 1-1 serves as both a welcome to the OracleAS Portal environment and an example of a typical OracleAS Portal page. Pages like this one that end users will display initially are commonly referred to as landing pages. This page contains a lot of OracleAS Portal-specific information that end users will not have any interest in, so it is not appropriate to have this page serve as your landing page. Chapter 12 describes how to change what the end users see as their landing page.
As you work with OracleAS Portal, you will become comfortable with the basic layout of an OracleAS Portal page (although, as a developer, you have great flexibility to make your OracleAS Portal pages look like whatever you want), and this welcome page contains all of the basic OracleAS Portal page elements.

Along the top of the page (above the first horizontal line) there is a page region called the banner. Although your portal pages do not have to contain a banner, it is included in all OracleAS Portal templates when it comes time to build and deploy your OracleAS Portal pages. You will find an image on the left, a title (Portal Builder), and links on the right. Below the links on the right is a tab that says Welcome. Tabs are similar to banners in that they are not required but are included as standard page elements in OracleAS Portal. Below the Welcome tab is the main part of the page.

**NOTE**

For those of you unfamiliar with graphical elements, tabs are standard page elements that look like the tabs that separate folders in your filing cabinet. Tabs can be used to provide easy navigation between a group of OracleAS Portal pages.

A key concept to understand when working with OracleAS Portal and the various OracleAS Portal elements is the one-to-many relationship. Oracle used this concept many times when constructing the OracleAS Portal environment. A one-to-many relationship describes one where there is one parent record and one or more child records. Child records cannot exist without a parent record, but a parent record may exist without any child records. As an example, say one of us walks into our local bookstore and purchases some books. The invoice might look like this:

```
Order #: 12345
Customer: 67890   Ostrowski, Chris
           215 Union Blvd
           Lakewood, CO 80228

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Oracle Application Server 10g Web Development, Ostrowski/Brown,</td>
</tr>
<tr>
<td></td>
<td>Oracle Press</td>
</tr>
<tr>
<td>1</td>
<td>Oracle9i Performance Tuning Tips &amp; Techniques, Niemiec, Oracle Press</td>
</tr>
<tr>
<td>1</td>
<td>Oracle PL/SQL Tips and Techniques, Trezzo, Oracle Press</td>
</tr>
</tbody>
</table>

Total: 179.97
Tax: 10.80
Total: 190.77

Payment Method: Mastercard xxxx-xxxx-xxxx-1234 Exp: 01/05
```
If the database used to store this information is normalized, there would be a
database record that stores information about the order (Order id = 12345) and a
line-items table that stores the line items that constitute this order (the three books).
In this case, the order record is the parent and the line items are the children. The
parent (the order) consists of one or more children (the line items). Line items
cannot exist independent of an order. It is also possible to split the payment due on
the order. Perhaps Chris has a gift certificate for $100. The payment to satisfy this
order might constitute the $100 gift certificate and $90.77 charged to his credit
card. Anticipating this possibility, the database is also designed with a one-to-many
relationship between the orders and payments tables.

TIP
“Normalized” and “normalization” are fancy words
for structuring your database so that there is as little
redundancy as possible. The objective is to increase
capacity by eliminating wasted storage.

For DBA-minded types out there, another example of the one-to-many relationship
is the one between tablespaces and datafiles in an Oracle database. A tablespace
(the parent) consists of one or more datafiles (children). A datafile cannot exist
independent of a tablespace.

In OracleAS Portal, developers will visit this one-to-many relationship many
times, and here is the first example of it. Every OracleAS Portal page, including the
Welcome page in Figure 1-1, is made up of regions. A region is nothing more than a
section of a page. Many aspects of content display are defined at the region level,
such as the width of the region or whether to display borders around the portlets in
a portlet region. Regions can also include one or more tabs. A region can be defined
to hold portlets, items, subpage links, and tabs, or it can exist as undefined
(undefined regions become defined when an OracleAS Portal object first gets placed
on them). You cannot add portlets to an item region, nor can you add items to a
portlet region. You cannot add anything to a subpage links region; these regions
automatically populate with links to subpages of the current page. You cannot add
anything other than a tab to a tab region; although you can configure the tab to
include, for example, rollover images. There is a one-to-many relationship between
pages and regions. In regions defined as portlet regions, there is also a many-to-
many relationship between regions and portlets. These relationships are shown in
Figure 1-2.
A real-world example

Orders

Line items

OracleAS Portal

OracleAS Portal Page

Region

OracleAS Portal Portlet

For every order in a database, there are 1-to-many line items associated with that order.

For every OracleAS Portal page, there are 1-to-many regions associated with that page. For each region, there are 1-to-many OracleAS portlets associated with that region.

**FIGURE 1-2  One-to-many relationship between pages, regions, and portlets**

**TIP**

Another way to think about regions is to picture each OracleAS Portal page as divided into a section of “Web real estate.” Each region exposes an OracleAS Portal object such as a portlet or an item and can be seen as a “container.” Every OracleAS Portal page consists of one or more regions.

The many-to-many relationship is slightly different from the one-to-many relationship discussed earlier. In this example, one region may have many portlets placed on it, but a portlet may be placed on one or many different regions on different pages. Item regions have the same one-to-many relationship: an item region may contain many items.

One of the many benefits of developing in OracleAS Portal is the ability to display pages with certain elements shown and others hidden, depending on the privileges of the user signing in to your portal. These changes are handled by OracleAS Portal automatically and require no additional programming from the developer. As an example, let’s log in to OracleAS Portal as if we were going to begin our OracleAS Portal development work. You can log in either by clicking the
small Login link on the top right of the page, or by clicking the Login link under the Login To OracleAS Portal portlet on the top left of the page. By default, a couple of OracleAS Portal logins are created for you automatically when you install OracleAS Portal. On the login screen, log in with the username of “portal” (without the quotes). The password will be the same as the password given when you or your administrator installed the Oracle Application Server 10g Application Server middle tier.

**CAUTION**
The portal user in OracleAS Portal is similar to the sys user in the database. It has every privilege within OracleAS Portal and should be used very sparingly. Just as when working as the sys user in the database, you should never create OracleAS Portal objects as the “portal” user. The portal user should be used for system administration purposes only.

After a successful login, you will see a page similar to Figure 1-3. The page looks similar to Figure 1-1, and in fact, the URL is exactly the same. But if you look closely, you’ll notice that there are many new elements on the page. On the top right of the page there are also many more links. Previously, only Home,
Builder, Help, and Logon were displayed. Now, Navigator, Edit, Customize, and Account Info are displayed and the Login link has been replaced with the Logout link. We also have three tabs along the top right: Welcome, Build, and Administer. Finally, the login portlet on the top left of the page has been replaced with a Quick Tips portlet. This is an example of how your OracleAS Portal pages can be designed to behave differently, depending on who is looking at them, with no additional programming whatsoever. When we first viewed the Welcome page, we had not authenticated ourselves to OracleAS Portal (authenticated is a fancy word for logging in). After we logged in, we became part of an Oracle Application Server group called the Authenticated Users group. As a member of that group, we saw a different version of the page. Our user login, portal, also gave us privileges to see other aspects of the page.
A Quick Tour of OracleAS Portal Pages

Because there are so many pages and tabs available to developers after they log in to OracleAS Portal, it is easy to become quickly overwhelmed. This section will walk through the basic OracleAS Portal pages and give a brief explanation of each so that later on, we can easily navigate among them as we visit them in detail.

Like most Oracle development tools, OracleAS Portal provides developers with many ways of performing the basic tasks of development and administration. None of these methods is the “right” way of doing things; they are simply different techniques for accomplishing your development goals that give developers the ability to choose which methods are suitable for them. Some of the methods require more steps than other methods, so after exploring the different methods, we will (generally) use those techniques requiring the fewest number of steps.

The Build Tab

As the portal user, you have all privileges in OracleAS Portal. As we have already discussed, OracleAS Portal pages display different elements according to the privileges granted to the user logging in, so many of these pages will look subtly different if you should log in as a different user. Clicking the Build tab in the top right will display a page similar to the one in Figure 1-4.

On this page there are four portlets:

- **Recent Objects portlet** This portlet allows developers and content administrators to quickly jump to one of the last five OracleAS Portal objects they’ve edited.

- **Developer News portlet** This portlet provides a link to Oracle’s Portal Development site (http://portalcenter.oracle.com/).

- **Page Groups portlet** This portlet allows developers to work with Page Groups, along with the banner page element discussed earlier. Page Groups are at the top of the hierarchy in Figure 1-2. Pages are children of Page Groups and cannot, therefore, exist without them. Every Page Group has a default root page, and any attribute defined for a Page Group automatically cascades down to its Pages (although many attributes can be overridden at the Page level).

- **Instant Portal portlet** Instant Portals are discussed in detail in Chapter 12.
The two portlets on the left-hand side of the page—Recent Objects and Developer News—give an example of OracleAS Portal’s security mechanisms. Both have a Personalize link in their title bars. These Personalize links allow end users to change certain characteristics of the portlet. Depending on the privileges granted to a particular user, the Personalize links will appear or be hidden. Any changes made to the portlet will be made for that user only. For example, if the portal user chooses to personalize the Recent Objects portlet to limit the list to only the three most recent objects (Figure 1-5), another user logging in will still see the five most recent objects in that portlet.

**FIGURE 1-4**  The Build tab displayed in OracleAS Portal
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

The Administer Tab

The third tab in Figure 1-4, Administer, is where OracleAS Portal administrators will spend most of their time. It has three subtabs along the left-hand side: Portal, Portlets, and Database. The Portal subtab (see Figure 1-6) has portlets that allow administrators to change the basic functionality of the OracleAS Portal as a whole. The Portlets subtab (see Figure 1-7) has portlets that allow administrators to display the Portlet Repository and define remote providers.

FIGURE 1-5 The Personalize Recent Objects portlet screen
Providers

In yet another example of the one-to-many relationship found throughout the OracleAS Portal product, a provider can be thought of as a way of grouping portlets together. Any attribute assigned to the provider will cascade down to the portlets in that provider (unless it’s overridden by the portlet). In addition, providers are also members of provider groups, which can also define attributes. The provider is the parent and the OracleAS Portal components (forms, reports, graphs, etc.) that can eventually become portlets are the children.

FIGURE 1-6  The Portal subtab in the Administer tab
The last subtab, Database (see Figure 1-8), allows administrators to create and modify both database objects and rows within a table. This functionality is limited to the infrastructure database.

Here is another example of OracleAS Portal’s security mechanism: Out of the box, OracleAS Portal defines Groups that have various privileges within OracleAS Portal (the portal user, of course, has all privileges). Becoming a member of any OracleAS Portal group automatically grants all privileges to that user. Three of the basic groups defined during the install of OracleAS Portal are

- PORTAL_ADMINISTRATORS
- PORTAL_DEVELOPERS
- PORTLET_PUBLISHERS
The OracleAS Portal elements on each of the tabs displayed along the top of the page will differ depending on what groups the OracleAS Portal user is assigned to. As an example, if a user is only a member of the PORTAL_DEVELOPERS group, nothing will be displayed in the Database subtab of the Administer tab. In general, these default groups were set up with the following privileges and for the following reasons:

**PORTAL_DEVELOPERS** Users in this group can create portlets but cannot place them on a page. In general, users in this group are usually more concerned with the functionality of the various portlets that will make up an OracleAS Portal site and less concerned with the look and feel of a site.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

- **PORTLET_PUBLISHERS**  Users in this group can take developed portlets and place them on a page as well as construct pages and page templates, but they cannot create new portlets. In general, users in this group are generally more concerned with the aesthetics of a portal site and less concerned with the code that makes up the various portlets to be placed on a site.

- **PORTLET_ADMINISTRATORS**  Users in this group have responsibilities across all facets of the site, including creation and deletion of users and groups, design and code modifications, and overall OracleAS Portal functionality. In general, users in this group have control over all aspects of the OracleAS Portal site.

You are not limited to these groups; you can create other groups that have distinct privileges and assign users to these groups. In some organizations, there may be a group of portlet users that is in charge of both designing portlets and placing them on pages. In this case, a new group with the privileges of both PORTAL_DEVELOPERS and PORTLET_PUBLISHERS can be created and users can be assigned to that (or users can just be assigned to both groups). If you have a small number of administrators and developers at your OracleAS Portal site, you may not even need groups at all, as you can assign these privileges directly to OracleAS Portal users.

**OracleAS Portal Navigator**

Up to this point, we have seen OracleAS Portal pages where developers can create users and groups, page groups, register providers, and change OracleAS Portal settings and database objects, among other things. What we haven’t seen is the ability to design pages that end users will typically view and interact with or to create Oracle Application Server portlets such as forms and reports. To do those things, we’ll need to explore another major piece of OracleAS Portal: the OracleAS Portal Navigator. Page developers can create/edit page groups, pages, styles, and templates, among other object types, directly from the build page. Oracle introduced this feature to make it easier for people to manage objects without relying on the OracleAS Portal Navigator so much.

The OracleAS Portal Navigator (see Figure 1-9) is where developers who will be using the wizards built into OracleAS Portal will spend most of their time. Its three tabs, Page Groups, Providers, and Database Objects allow developers to create, modify, and delete all of the OracleAS Portal objects that make up a portal. Again, only certain tabs and certain options within those tabs will be available to you as
the OracleAS Portal developer, depending on your privileges. You can exit the Navigator at any time by clicking the Builder link on the top right of the page.

**The Page Groups Tab**

The Page Groups tab allows the creation of all page elements. On this tab, developers can perform the following actions:

- **Create a new Page Group**  This action, which also automatically creates the root page of that group, is used to define a group that will incorporate pages that make up your portal or a section of your portal. As an example, an organization may have a set of OracleAS Portal pages with a certain look for a corporate office, another set for external suppliers, and other sets
of pages for various satellite locations. Any attributes applied at the Page Group level cascade down to all subpages (unless they're overridden at the subpage level).

- **Edit properties of the Page Group**  This action allows developers to define attributes of the Page Group such as the total amount of disk space used for items placed on pages within this page group, whether privileged users can alter the pages within the Page Group, what types of pages can exist within the Page Group, how items are to be displayed and versioned, if language translations are to be made available, and which users can perform which actions against pages in the page group.

- **Create new subpages**  Every page group has a root page; other pages can be created as subpages to the root page. A breadcrumb menu, showing where the user is in the hierarchy of pages, is generated automatically.

- **Edit the root page or any of the subpages**  This is where users typically assigned to the PORTLET_PUBLISHERS group will design pages and place portlets and items on them.

- **Create, modify, and apply Templates**  These are used for enforcing a particular layout, style, set of privileges, and content for multiple pages.

- **Create and modify Categories**  The purpose of a category is to enable users to quickly display a listing of a particular type of content. Categories answer the question, “What is this item or page?” and are used to classify content. For every item or page that you create, you can assign it to one category.

- **Create and modify Navigation Pages**  A navigation page is a special type of page that can be included on other pages to provide a consistent set of navigational elements. A typical navigation page might contain a logo, the page title, a login link, and a link to the home page. Navigation pages differ from other pages in that they are excluded from searches and bulk actions performed on pages in the page group. Additionally, they have their own node in the Navigator (Navigation Pages). Although you can add any item or portlet to a navigation page, and even divide your navigation page into several regions, you should bear in mind that the idea is to add navigation pages to other pages.

- **Create and modify Perspectives**  The purpose of perspectives is to enable users to quickly display a listing of content that is pertinent to them. Perspectives often answer the question, “Who will be interested in this item or page?” and are used to further classify content by a cross-category grouping. When an item is added to a page in the page group, it can be
assigned to multiple perspectives. To find all of the available content, other users can then search on this perspective to see a listing of the pages and items that are part of it. Since perspectives have the potential to be large, developers or content managers can also create a hierarchy of perspectives.

- **Create and modify Styles**  A style controls the colors and fonts used by pages and all the tabs, portlets, and items displayed within them.

- **Create and modify Attributes**  There are two types of attributes:

  - **Content attributes**  These are associated with item types and page types, and they store information about an item or page. Administrators can create their own item types and page types and specify what information they want users to supply by choosing which attributes to include. In addition, page group administrators can create their own attributes for containing extra information. The following table lists built-in content attributes:

    | Attribute     | Description |
    |---------------|-------------|
    | Author        | The name of the author of the item. |
    | Category      | The name of the category to which the item or page belongs. |
    | Description   | A short text description of the item or page. |
    | Display Name  | The display name of the item or page. |
    | Display Option| Information about how the item or portlet should be displayed: |
    |               | ■ Item Displayed Directly In Page Area |
    |               | ■ Link That Displays Item In Full Browser Window |
    |               | ■ Link That Displays Item In Full Browser Window |
    | Enable Item Check-Out | Information about whether or not the item can be checked out and checked in. This provides document control, allowing groups of users to edit items and not overwrite each other’s work. Users cannot edit items that are checked out by another user. |
    | Expiration Period | Information about how long an item should be displayed on a page. |
    | Image         | The image associated with the item or page. |
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Alignment</td>
<td>Information about where the item image should appear in the page.</td>
</tr>
<tr>
<td>Keywords</td>
<td>Keywords that describe the content or purpose of the item or page. When a user performs a search, the user’s search criteria are compared to the keywords to find a match.</td>
</tr>
<tr>
<td>Perspectives</td>
<td>The names of the perspectives associated with the item or page.</td>
</tr>
<tr>
<td>Publish Date</td>
<td>The date (in the format, DD-MON-YYYY HH12:MI PM) when the item should start being displayed to users.</td>
</tr>
<tr>
<td>Rollover Image</td>
<td>The second image associated with the item or page. This image is displayed whenever a user moves the mouse cursor over the original image on a navigation bar or tab.</td>
</tr>
</tbody>
</table>

Display attributes These are associated with regions and display information about an item or portlet, such as the author, display name, and creation date. Page designers can choose which attributes to display in a region. Some content attributes, such as author and description, are also display attributes. The following table lists built-in display attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associated Functions</td>
<td>Links to functions associated with the item if the item is of a custom item type that includes function calls.</td>
</tr>
<tr>
<td>Author</td>
<td>The name of the author of the item.</td>
</tr>
<tr>
<td>Category</td>
<td>The name of the category to which the item or portlet belongs.</td>
</tr>
<tr>
<td>Create Date</td>
<td>The date when the item or portlet was added to the page.</td>
</tr>
<tr>
<td>Creator</td>
<td>The user name of the user who added the item or portlet to the page.</td>
</tr>
<tr>
<td>Date Updated</td>
<td>The date when the item or portlet was last updated.</td>
</tr>
<tr>
<td>Description</td>
<td>The short text description of the item or portlet.</td>
</tr>
<tr>
<td>Display Name</td>
<td>The display name of the item or portlet.</td>
</tr>
</tbody>
</table>
### Attribute Description

**Display Name Link**
The display name of the item or portlet as a link pointing to the item or portlet content.

**Display Name And Image Link**
The display name and image of the item or portlet as links pointing to the item or portlet content. If the item has both a display name and an image, these will appear next to each other. If the item does not have an associated image, only the display name will appear.

**Document Size**
The size of the uploaded file.

**Expire Date**
The date (in the format, DD-MON-YYYY HH12:MI PM) when the item is due to expire.

**Gist**
The Gist icon next to items. Users can click this icon to display an overview of the item created by Oracle Text.

**Help URL**
The help icon next to the portlet item. Users can click this icon to display help for the portlet.

**Image**
The image associated with the item. If the item does not have an associated image, nothing is displayed.

**Image Link**
The image associated with the item as a link pointing to the item content. If the item does not have an associated image, nothing is displayed.

**Image Or Display Name Link**
The image associated with the item as a link pointing to the item content. If the item does not have an associated image, the display name is displayed instead.

**Image URL**
The image icon next to the portlet item. Users can click this icon to display a preview of the portlet.

**Item Content**
The content of the item.

**Keywords**
The keywords associated with the item or portlet.

**Last Updated By**
The user name of the user who last updated the item or portlet.

**Mime Type Image**
The image associated with the MIME type of the uploaded file.
### Attribute Description

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Item Indicator</td>
<td>The New icon next to a new item or portlet. Users can click this icon to display a list of all new content in the page group.</td>
</tr>
<tr>
<td>Perspectives</td>
<td>The names of the perspectives associated with the item or portlet.</td>
</tr>
<tr>
<td>Portlet Content</td>
<td>The portlet itself if the Item Displayed Directly In Page Area display option is selected.</td>
</tr>
<tr>
<td>Portlet ID</td>
<td>The ID of the portlet.</td>
</tr>
<tr>
<td>Portlet Name</td>
<td>The name of the portlet.</td>
</tr>
<tr>
<td>Property Sheet</td>
<td>The Property Sheet icon next to items and portlets. Users can click this icon to view the properties of an item or portlet.</td>
</tr>
<tr>
<td>Provider ID</td>
<td>The ID of the provider.</td>
</tr>
<tr>
<td>Provider Name</td>
<td>The name of the provider.</td>
</tr>
<tr>
<td>Publish Date</td>
<td>The date when the item is published on the page (i.e., when the item is visible to users in View mode).</td>
</tr>
<tr>
<td>Subscribe</td>
<td>The Subscribe or Unsubscribe icon next to items. Users can click this icon to subscribe to an item and be notified, via the Notifications portlet, when it is updated.</td>
</tr>
<tr>
<td>Themes</td>
<td>The Themes icon next to items. Users can click this icon to display the nouns and verbs appearing most often within an item.</td>
</tr>
<tr>
<td>Translations</td>
<td>A list of the languages in which the item or portlet is available.</td>
</tr>
<tr>
<td>Updated Item Indicator</td>
<td>The Recently Updated icon next to a recently updated item or portlet. Users can click this icon to display a list of all recently updated content in the page group.</td>
</tr>
<tr>
<td>Versions</td>
<td>The Versions icon next to items that have multiple versions. Users can click this icon to view other versions of the item.</td>
</tr>
<tr>
<td>View As HTML</td>
<td>The View As HTML icon next to items. Users can click this icon to view an HTML version of an item.</td>
</tr>
</tbody>
</table>
Create and modify Page Types  Page types define the contents of a page and the information that is stored about a page. The information stored about a page is determined by the attributes of the page type. There are five base page types included with OracleAS Portal:

- **Standard**  Displays items and portlets
- **URL**  Displays the contents of a particular URL
- **Mobile**  Displays item and portlets in a hierarchical tree structure for viewing on a mobile device
- **PL/SQL**  Displays the results of executing PL/SQL code
- **JSP**  Displays the results of executing a Java Server Page (JSP)

Custom Page Types
In addition to these basic page types, developers and page designers can also create custom page types. Items are one of the basic components of a portal page. Items in OracleAS Portal are based on item types. Item types define the contents of an item and the information that is stored about an item. The information stored about an item is determined by the attributes of the item type. There are two kinds of item types: base item types and navigation item types.

Base Item Types  Base items can be broken down further into these subtypes:

- **Content item types**  These allow users to add content (for example, images, documents, or text) to a page. Base content item types are not available for users to add to pages. OracleAS Portal provides extended item types (listed next) that are based on the base content items. Oracle offers the following base content item types:
  - **Base File**  Uploads a file and stores it in the page group
  - **Base Image Map**  Uploads an image and allows the contributor to identify areas within the image that users can click to go to different URLs
  - **Base Image**  Uploads an image and stores it in the page group
  - **Base PL/SQL**  Executes PL/SQL code and displays the results
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

- **Base Page Link**  Links to another page in the page group
- **Base Text**  Displays text (up to 32KB)
- **Base URL**  Links to another Web page, Web site, or document

**Navigation Item Types**  These allow users to add navigational elements (for example, a login/logout link, basic search box, or list of objects) to a page. The following are the base navigation item types provided by Oracle:

- **Portal Smart Link**  Adds a smart link (and associated image) to the page. A smart link is a link that users can click to access areas of the OracleAS Portal quickly, such as Account Information, Advanced Search, Contact Information, Help, and Home.
- **Login/Logout Link**  Adds links and/or icons to the page that users can click to log in to or log out of the portal.
- **Basic Search Box**  Adds a basic search box (and associated image) to the page in which users can enter search criteria. Users can specify whether users of the search box can search all page groups or only the page group specified.
- **List of Objects**  Adds a list of objects (pages, categories, and perspectives) that users specify to the page. Users can choose to display this list as a drop-down list or as links (with or without associated images).
- **Portal Smart Text**  Adds smart text, such as the current date, current user, or current page, to the page.
- **Object Map Link**  Adds a map of objects available in the portal.
- **Page Path**  Adds the page path to the page. Users can choose the number of levels for the path, and the character that separates the path levels.
- **Page Function**  Adds a page function to the page. If there are no page functions associated with the current page, this item type is not displayed.

**Extended Item Types**  These types are available to users to add to pages:

- File and Simple File
- Simple Image
- Image and Simple Image Map
- PL/SQL and Simple PL/SQL
Chapter 3 and 4 discuss OracleAS Portal page design in detail. Chapter 5 discusses content management and the various functions associated with types, items, and item attributes.

The Providers Tab
As we mentioned earlier, a provider can be thought of as a way of grouping portlets together. On the Providers tab of the Navigator in Figure 1-10, you can see that there are three categories of providers available: Locally Built Providers, Registered Providers, and Provider Groups.

Locally Built Providers This is where developers will define providers when they are ready to build OracleAS Portal objects such as OracleAS Portal forms, reports, and graphs. This chapter focuses on using the OracleAS Portal wizards and therefore deals mainly with Locally Built Providers.

Registered Providers This is where providers built outside of the OracleAS Portal are. Registered providers can include the following:

- **Database providers** A database provider is one that is written as a PL/SQL package. Use database providers for creating PL/SQL portlets that reside in the database. PL/SQL portlets are implemented as stored procedures and executed in the database. They can be written in PL/SQL or Java Stored Procedures wrapped in PL/SQL. Use PL/SQL portlets whenever your portlets require significant database interaction or when the development team has Oracle experience.

- **Web providers** A Web provider is one that is written as a Web application. It is installed and hosted on a Web server and is remote from the portal. A portlet exposed as a Web provider can be developed in any Web language. The portal communicates to the Web provider using the HTTP protocol. There are several benefits when developing portlets and exposing them as Web providers:
  - Leverage existing Web application code to create portlets
  - Manage outside of OracleAS Portal
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

- Provide hosted servers for OracleAS Portal users
- Create portlets using any existing Web language

**NOTE**
This chapter deals mainly with locally built providers. Chapters 7 and 8, however, discuss Web providers.

**Provider Groups**  A provider group is a mechanism for organizing and simplifying the registration of providers. Provider groups often define a group of providers that share a common feature, such as belonging to the same organization, or providing similar content or functions.

Part III of this book discusses providers in more detail.

**FIGURE 1-10** The Providers tab of the OracleAS Portal Navigator
The Database Objects Tab

The Database Objects tab allows OracleAS Portal users with the necessary privileges to manipulate database objects in the infrastructure database (see Figure 1-11). Some of the actions that can be performed include:

- Creating new schemas
- Creating and modifying these database objects: tables, views, procedures, functions, packages, sequences, synonyms, indexes, triggers, database links, Java objects, and scripts
- Querying rows in tables and views
- Modifying columns and column attributes
- Modifying rows in tables and views
- Viewing table constraints and column attributes

FIGURE 1-11 The Database Objects tab of the OracleAS Portal Navigator
The Help System

Last, but certainly not least, is the OracleAS Portal Help system (see Figure 1-12). Oracle has greatly improved the help system included with OracleAS Portal, including context-sensitive help for most (but not all) screens in OracleAS Portal version 10.1.4. The Search tab on the help pages is invaluable, as it allows developers to search through all of the OracleAS Portal documentation in one place.

FIGURE 1-12  The OracleAS Portal help system
Creating an OracleAS Portal User

The first order of business we should take care of is the creation of a user to develop OracleAS Portal objects. The creation of initial users is one of the main reasons you would ever log in to your portal as the portal user. Assuming you are still logged in as the portal user, click the Administer tab on the top right of the Welcome page. Make sure the Portal subtab is selected on the top left of the page and click the Create New Users link on the top right of the page. After you select that link, your browser will be directed to a long URL that starts something like this:

http://<infrastructure server>:<infrastructure port>/oiddas/ui/oracle/ldap/das/user/AppCreateUserInfoAdmin ...

The examples in the chapter have been taken from a server named oski-2k3 with both the infrastructure (port 7777) and the middle tier (port 80) installed on it. On this machine the re-directed URL begins like this:

http://oski-2k3:7777/oiddas/ui/oracle/ldap/das/user/AppCreateUserInfoAdmin ...

It is important to note this for the following reason: when creating OracleAS Portal users, we are creating users in Oracle’s implementation of the Lightweight Directory Access Protocol (LDAP) standard, Oracle Internet Directory. Creating an OracleAS Portal user does not create a user in the Infrastructure database.

LDAP (Lightweight Directory Access Protocol) is a set of protocols for accessing information directories. LDAP is based on the standards contained within the X.500 standard but is significantly simpler. And unlike X.500, LDAP supports TCP/IP, which is necessary for any type of Internet access. LDAP makes it possible for almost any application running on virtually any computer platform to obtain directory information, such as e-mail addresses and public keys. Because LDAP is an open protocol, applications need not worry about the type of server hosting the directory.

TIP
You can also create and edit Oracle Internet Directory users at any time by going to http://<infrastructure server>:<infrastructure port>/oiddas. The root user for Oracle Internet Directory is orcladmin, and the password will be the same one assigned to the infrastructure instance during installation of the infrastructure.
We will now create users to handle the various tasks associated with development and administration of our portal. After selecting the Create New Users link, you should see a page similar to the one in Figure 1-13. On this page, you will create a new user and assign privileges that will allow that user to create OracleAS Portal components. Fill in the required fields for a user you will use for OracleAS Portal development and click the Roles Assignment link (see Figure 1-14).

![FIGURE 1-13 The Create User page of the Oracle Internet Directory](image)

Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies
Figure 1-14 lists the default roles that are provided when OracleAS Portal is installed. You are not limited to these default roles; you can create your own. For now, pay attention to the roles listed earlier in this chapter, namely PORTAL_DEVELOPERS, PORTLET_PUBLISHERS, and PORTLET_ADMINISTRATORS. Click the check box next to PORTAL_DEVELOPERS as in Figure 1-14. Click the Return To Top link and then click the Submit link to create the new user. In another browser, log in to OracleAS Portal with this new user’s name and password.

Clicking the Administration tab and then the Database tab will not reveal any portlets, since this user does not have administration privileges. This user does, however, have the ability to create OracleAS Portal components. Click the Navigator link on the top right of the page to be taken to the OracleAS Portal Navigator.
all OracleAS Portal components must be associated with a provider, select the Providers tab, and then select the Locally Built Providers link on that page. The Locally Built Providers page is displayed (see Figure 1-15), but it contains no links to create a new provider. This is one of the security features that frustrate many new OracleAS Portal developers. Instead of presenting a link or option that brings you to a page or screen explaining the fact that the OracleAS Portal user does not have sufficient privileges to perform an action, the link or option simply is not displayed on the screen. The user we have created has been granted the privileges in the PORTAL_DEVELOPERS group, which gives us the ability to create OracleAS Portal components, but not Oracle Application Server Providers. Why?

FIGURE 1-15 The Locally Built Providers page
The OracleAS Portal wizards that are used to generate forms, reports, and graphs (among other OracleAS Portal components) are really sophisticated code generators. While you have the ability to embed JavaScript code with your OracleAS Portal Forms and Reports, the majority of code that is generated by the OracleAS Portal wizards will be PL/SQL code. This PL/SQL code needs to be stored somewhere: it is stored in the infrastructure database. When an OracleAS Portal Provider is created, one of the first things we will need to specify is a database schema that this provider is associated with. This database schema is the one that will be used to hold the generated PL/SQL code that makes up the various OracleAS Portal objects that we generate from the wizards. In order to create an Oracle Application Server Provider, we must have access to the database (the database installed in the Oracle Application Server infrastructure) to specify which schema we want to use. In most cases, you would not want to give that privilege to an OracleAS Portal developer, which is why our user does not have the ability to create a new provider.

What if we want to give that user the ability to create a provider? There are a couple of ways to do it. We can either grant a provider privilege explicitly to that user, or we can make that user a member of a group with the necessary provider privilege so that the user implicitly gets the necessary privilege. Let’s look at explicitly granting the privilege first.

Log in to Oracle Portal Provider Portal as a user with administration privileges (you can use the portal user we’ve been using in this chapter). Click the Administer tab. On the right-hand side of the screen, you’ll see four portlets: User, Portal User Profile, Group, and Portal Group Profile. To change the OracleAS Portal privileges for a user, you may think you need to go into the User portlet, but you would be incorrect. The User portlet is only used for things like the OracleAS Portal user’s personal information (username, password, group memberships, etc.). To edit an OracleAS Portal user’s privileges, we will use the Portal User Profile portlet. Click the small icon between the Name: field and the Edit button to bring up a list of OracleAS Portal users defined on your system. Select the user that was just created and click Select to close the selection window. Click the Edit button to bring up the Edit Portal User Profile screen. Click the Privileges tab to display privileges for that user (see Figure 1-16).

It looks like this user does not have any privileges to do anything in our portal, but remember that the user inherits the privileges of any group that user is assigned to. In this case, no explicit privileges have been assigned to this user, but he has implicitly inherited the privileges from the PORTAL_DEVELOPERS group. Click the Builder link on the top right of the screen, and then click the small icon between the Name: field and the Edit button in the Portal Group Profile portlet at the bottom of the page. Select the PORTAL_DEVELOPERS group, click Edit, and select the Privileges page. As you can see in Figure 1-17, members of the PORTAL_DEVELOPERS group have the ability to create Oracle Application Server Providers.
So what’s going on here? We know that the user we have just created is a member of the PORTAL_DEVELOPERS group and that the PORTAL_DEVELOPERS group has the ability to create providers. Why doesn’t the Create Provider link show up when we log in as that user? Remember, we said that providers must be associated with a database schema so that the PL/SQL packages that are generated from the various OracleAS Portal wizards can be stored. The OracleAS Portal user must have privileges on those schemas so that these packages can be created. We must grant an additional privilege to this OracleAS Portal user so that the user can create providers. On the Administration tab and on the Portal subtab, type the name of your OracleAS Portal user in the name field of the Portal User Profile portlet and click Edit. On the Privileges tab, select the drop-down box next to All Schemas. You’ll see six options: Create, View Data, Insert Data, Modify Data, Manage, and None. The order in which they are listed is a little confusing: moving from the
bottom of the list (Create) up, the privileges encompass more and more capabilities, except for the top-most selection (None), which revokes all privileges. It would seem to make more sense to put None at the bottom of the list, but it is at the top for all drop-down boxes on this page.

The Create option gives the OracleAS Portal user the ability to create a new schema in the infrastructure database, but nothing else. View Data gives the OracleAS Portal user the ability to create a schema and query the data in that schema, but no privileges to add or insert data. Insert Data has all of the privileges of those below it, plus the ability to insert data into the schema. Granting our OracleAS Portal user any of the three privileges we’ve mentioned will not give the user the ability to create an OracleAS Portal Provider. Providers not only need the ability to view and insert data as various elements of the OracleAS Portal
components are created, but also need the ability to modify data in that schema as OracleAS Portal components are changed and updated. We need a privilege higher than Insert Data before our OracleAS Portal user will have the ability to create providers. The last two options—Modify Data, which has all of the privileges below it plus the ability to actually modify data in the schemas, and Manage, which has the ability not only to perform Data Manipulation Language (DML) statements like insert, update, etc., but also to perform Data Definition Language (DDL) statements like “Create Index”—give the necessary privileges for our user to create Oracle Application Server Providers. For now, as an example, grant the user the Manage privilege and click OK. Log back in as the new OracleAS Portal user, click Navigator, then the Providers tab, and then the Locally Built Providers link. The page should look similar to before, except with the addition of the Create New... Database Provider link at the top of the page (see Figure 1-18).

FIGURE 1-18 The Locally Built Providers page of the Portal Navigator with the Create New Database Provider link available
There are two types of providers—database providers and Web providers—in OracleAS Portal. Database providers are those packages written as PL/SQL packages. They’re used for creating PL/SQL portlets that reside in the database and are implemented as stored procedures and executed in the database. They can be written in PL/SQL or Java Stored Procedures wrapped in PL/SQL. You use PL/SQL portlets whenever your portlets require significant database interaction or when the development team has Oracle experience. Web providers are those applications written as Web applications. They are installed and hosted on a Web server and are remote from the portal. A portlet exposed as a Web provider can be developed in any Web language. The portal communicates to the Web provider using the HTTP protocol. There are several benefits when developing portlets and exposing them as Web providers: You can leverage existing Web application code to create portlets, manage outside of OracleAS Portal, provide hosted servers for OracleAS Portal users, and create portlets using any existing Web language. Web providers use Simple Object Access Protocol (SOAP) to communicate with the Portal. Web providers, such as OC4J (along with the PDK), can also be installed locally on the Oracle Application Server. This chapter deals primarily with database providers, but the majority of this book deals with Web providers.

The example just discussed illustrates how tightly security is built into OracleAS Portal. It also serves to demonstrate that it is not always intuitively obvious how and where to make the changes necessary to grant access to your OracleAS Portal users.

Now that we have the ability to create providers, let’s go ahead and create one to see what options are available to us. As the OracleAS Portal user created in this chapter, click the Navigator link on the top right of the screen, then the Providers tab, and then the Database Provider link on the top left of the screen. You should see a screen similar to the one in Figure 1-19.

The first two fields are self-explanatory: the internal name of the application (which cannot contain any spaces or special characters) and the display name of the application (which can contain spaces and special characters and will be what is displayed in the OracleAS Portal Navigator). The third field is more troublesome in this example. It requires us to specify which schema in the infrastructure database we will use to store our PL/SQL packages. As you can see in Figure 1-19, no schemas are available to us, preventing us from continuing. What’s happening here?

When we were modifying this user earlier, we gave the OracleAS Portal user the ability to modify data in any schema in our infrastructure database. When we use the OracleAS Portal wizards to create forms, reports, graphs, etc., we will need the ability to create various database objects in the schema; granting Modify Data privileges doesn’t give us the ability to actually create any new database objects, so no database schemas are available to us.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

So the answer is simple, right? Go back and grant Execute privileges to our OracleAS Portal development user. That will give the OracleAS Portal developer the necessary privileges to create a provider, but that may not be such a great idea, either. By granting that privilege, an OracleAS Portal developer can do anything to any of the schemas in the infrastructure database. The potential for disaster, unintentional or on purpose, is great in this scenario. A better solution would be to define a schema to hold Oracle Application Server Portal–generated PL/SQL packages and grant privileges on that schema to the necessary OracleAS Portal developer(s).

You can use OracleAS Portal to create a schema in the database. Log in as a user with administration privileges (like the portal user) and select the Navigator link. Select the Database Objects link and then the Create New... Schema link. Fill in the necessary fields and make sure the Use This Schema For Portal Users check box is selected.
selected. Click Create. Back on the Navigator page, select the Grant Access link next to the schema you just created. You should see a page similar to the one in Figure 1-20. On this page, you can grant privileges to OracleAS Portal user on the selected database schema. Much as with the Privileges tab discussed earlier, the privileges you can select on this page (view, insert, modify, and manage) are in order from least powerful (view) to most powerful (manage). Granting view, insert, or modify will not grant enough privileges for the OracleAS Portal user to use this schema to begin creating OracleAS Portal components for a provider. You must grant the Manage privilege for your OracleAS Portal developer to use this schema. Click Add before clicking OK to close the page.

**FIGURE 1-20  The Grant Access page**
Logging back in to OracleAS Portal as the developer user, you can now create a new provider using the schema that has been granted the necessary privileges (see Figure 1-21).

The granting of Manage privileges on the portal_dev schema in the preceding example is all the OracleAS Portal developer user needs to create providers (along with, of course, the Create Provider privilege implicitly granted by being a member of the PORTAL_DEVELOPERS group). There is no need to grant the individual user the Manage privilege for all users that we performed earlier (just before Figure 1-18). To keep security tight in your database, it’s best to go back and revoke that privilege.
Portlets

What is a portlet? Oracle’s definition is that they are “reusable building blocks for easily publishing information and applications.” You can think of a portlet as a small application that performs a specific function. Portlets are then placed and arranged on a page so that the end user can interact with them. Portlets can be forms, reports, graphs, links to other Web sites, ad hoc query tools—the list goes on and on. All portlets come from a data source registered within OracleAS Portal, called a portlet provider. You can use OracleAS Portal’s wizards to easily create reports, forms, charts, and other types of dynamic components or even publish pages, navigation pages, and other OracleAS Portal components as portlets. You can also build components with your own tools and integrate them through OracleAS Portal’s APIs, available in the Portal Developer Kit (PDK).

One of the most difficult, yet ultimately beneficial, concepts for many beginning OracleAS Portal developers to grasp is the fact that the OracleAS Portal development environment is itself an OracleAS Portal application. All forms and pages that developers use to generate OracleAS Portal components are OracleAS Portal elements themselves, stored in the OracleAS Portal repository. This is beneficial to developers because they can work with a well-designed portal environment and understand the basics of OracleAS Portal development, architecture, and navigation before attempting to build and deploy their first portal pages.

NOTE
The wizards, even though they look just like OracleAS Portal pages, are technically not really OracleAS Portal objects. They are PL/SQL-coded solutions, and the source code is not published.

This chapter is designed to show the architecture of OracleAS Portal and to provide a road map so that developers and administrators can find their way around OracleAS Portal quickly. Even though OracleAS Portal is a true declarative development environment that requires little, if any, coding for a complete application, it is still beneficial to define and explore the structure of OracleAS Portal and how applications and portals are constructed. Given the fact that OracleAS Portal is a true Web-based development environment, you are probably anxious to jump in and start creating OracleAS Portal portlets and applications immediately; some of the material in this chapter may appear dry in that context, but mastering the basics of OracleAS Portal navigation will save you much time as your development efforts move forward.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

Types of Portlets
OracleAS Portal supports numerous types of portlets:

- **Java** A portlet implemented using Java technologies. Part III of this book focuses on developing these types of portlets using Oracle JDeveloper.

- **PL/SQL** A portlet implemented using PL/SQL technologies. PL/SQL portlets can be developed from scratch but are most commonly developed using the OracleAS Portal wizards. This chapter discusses this technique, but the majority of this book focuses on developing portlets using Java technologies.

- **OmniPortlet** A Web provider that provides portlets that can display spreadsheet, XML, and Web Service data as tabular, chart, news, bullet, and form layouts. OmniPortlet is discussed in Chapter 6.

- **Web Clipping** A feature that enables page designers to collect Web content into a single centralized portal. It can be used to consolidate content from hundreds of different Web sites scattered throughout a large organization. Web Clipping is also discussed in Chapter 6.

- **Seeded portlets** Numerous portlets are available to developers when OracleAS Portal is installed. In Chapter 3, a section called “Seeded Portlets” discusses these portlets.

OracleAS Portal Content
OracleAS Portal contains sophisticated methods and programs to store, display, and manage content, giving administrators almost limitless ways to administer content on their portals. The Content Management SDK gives administrators and developers the ability to automate many of the tasks of content management. Content Management and the Content Management SDK are discussed in detail in Chapter 5.

TIP

A good place to start our discussion of OracleAS Portal content would be the definition of content. As we have seen, OracleAS Portal allows developers to build components such as forms, reports, charts, calendars, etc. But what if you have
information that you wish to display on your portal that does not fit into one of these component types; for example, a Microsoft Word document or an Adobe PDF file? The developers of OracleAS Portal were smart enough to design it so that both types of objects can be displayed on your Portal: portlets, which can query Oracle databases and display that data in a form, report, calendar, etc.; and items, which can display content such as word processor files, spreadsheets, or images. Within the context of OracleAS Portal, content can be defined as any piece of information that is to be displayed on a portal that does not fit into the traditional interface of an OracleAS Portal component such as a form or report. There are two basic types of items that can be placed on a page: Content Item Types and Built-in Navigation Types.

Content Item Types
The following is a list of Content Item Types you can place on your OracleAS Portal pages:

- **File**  This option allows you to place a file on your portal and is the most common option selected when placing content on your portal. When a file is selected, it is converted to a binary large object and stored in the infrastructure database automatically. You also have the option of displaying a Simple File, which does not prompt you for advanced content attributes such as publish or expiration dates (advanced content attributes are discussed in Chapter 5).

- **Text**   This option displays a WYSIWYG (What You See Is What You Get) editor that allows you to enter text for your content area. Just as with the File option, there is a Simple Text option.

- **URL**    This option allows you to place a URL in your content area. Alternatively, the editor displayed in the preceding text option allows you to create links in your text area. As with the File and Text options, there is also a Simple URL option.

- **Page Links**  This option allows you to place links in your content area to other OracleAS Portal pages. There is also a Simple Link option.

- **Images**  This option allows you to place an image in your content area. There is also a Simple Image option.

- **Zip Files**  This option allows you to upload a Zip (compressed) file to your content area. The only difference between this and a File is the existence of
a link titled Unzip next to the item when displayed. Clicking the Unzip link gives end users the ability to unzip and store whatever is in the Zip file on a page in the portal.

- **PL/SQL**  This option allows you to store a PL/SQL code fragment as a content item in the content area. Clicking this link will execute the PL/SQL code.

- **Oracle Reports**  This option allows developers to embed an Oracle Report (not to be confused with an OracleAS Portal Report) in a content area. Clicking the link will display the Oracle Report on the Web.

**Built-in Navigation Types**
The following is a list of Built-In Navigation Types you can place on your OracleAS Portal pages:

- **Portal Smart Links**  This option allows you to place various links in the content area to perform actions such as edit user account information, edit the page, display a help menu, take the user to his or her personal page, refresh the page, etc.

- **Login/Logout Links**  This option places a Logout link on the content area page.

- **Basic Search Box**  Two types of searches are available to end users in OracleAS Portal: A basic search allows users to search through their portals without reducing the information returned by category or perspective. An advanced search gives end users the ability to use various methods of reducing information returned by their search as well as utilize advanced features such as Boolean operators. The Basic Search box places a Search field on the content area, allowing end users to perform a basic search through content on the portal.

**NOTE**

For now, understand that categories and perspectives are ways of organizing content in your portal. Categories organize the content according to what the content is. Perspectives organize the content according to who might be interested in it.
List Of Objects  This option allows you to return a set of OracleAS Portal content that meets certain criteria. You can return a drop-down list or a set of links that point to a page group, a perspective, or a category.

Portal Smart Text  This option can be used to display the current date, user, or page.

Object Map Link  This option can be used to create a link that will display a hierarchical map of pages and subpages when clicked by an end user.

Page Path  This option creates a breadcrumb menu on the page that allows end users to see where they are in the portal site and gives them the ability to navigate through levels quickly. Placing a Page Path navigation link on a root page has no effect; it is only useful on subpages.

TIP  Not all of these types are available for pages by default. Some of the items will have to be enabled for a page group. To add these types, select the Properties link next to Page Group on the top left of the page editor. Select the Configure tab, and then click the Edit link in the Content Type and Classification section of the page. Add the Item Types in the Item Types section of the page and click OK.

Declarative Development
The term declarative development refers to the process of using the OracleAS Portal wizards to build and deploy your portlets. While this method has been available to Portal developers since the inception of Oracle Portal (known in its first public iteration as “WebDB”), Oracle has focused less on enhancing the OracleAS Portal wizards and more on features and functionality related to Web-based providers (Java, etc.). In the 10.1.4 version of the OracleAS Portal Developer’s Guide (http://download-east.oracle.com/docs/cd/B14099_19/portal.1014/b14135/toc.htm), the section on declarative development has been relegated to the appendix.

What does this mean for OracleAS Portal developers? Oracle will continue to support the OracleAS Portal wizards but will probably not add much functionality to them in the future. To explore the advanced features of OracleAS Portal development, the developer should concentrate on Web-based providers using a tool like Oracle JDeveloper. As such, this chapter briefly discusses the OracleAS Portal wizards, but all other development-based chapters in this book will focus on using Oracle JDeveloper to create Web Provider–based portlets.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

The OracleAS Portal Wizards
The OracleAS Portal wizards allow developers to create, with a minimum amount of hand-written code, portlets that can be deployed to the OracleAS Portal easily and securely. While this functionality gives beginning developers an easy way to achieve quick success, any wizard-based development tool, by its very nature, will have its limitations. While the advanced page design features give developers and publishers great functionality over how their applications look and feel, end users are demanding more and more functionality from their Web-based applications every day. It is difficult to create a sophisticated application that will satisfy a majority of users in a production environment using the OracleAS Portal wizards alone.

Common Features of all Wizards
All of the wizards have a common look and feel (many of the wizards for the different OracleAS Portal components have the exact same wizard pages), allowing developers to create and modify portlets quickly and easily. All of the wizards use a standard notation of “Step X of Y” along the top of the screen to show developers how far they are along in the development process for that particular portal component. Beyond the first couple of screens, where the type of portlet is selected and named, it is possible to change almost any facet of the portlet, either during the initial run of the wizard or by modifying the portlet after it has been generated.

For every portlet created using the OracleAS Portal wizards, there is a Manage screen that allows developers to modify the portlet, set privileges for the portlet, and view the portlet before it gets placed on an OracleAS Portal page. The Manage page also allows developers to export the portlet (useful for moving a portlet from a development environment to a production environment). The Manage page also gives you the ability to view the code that the wizard has generated and see the call interface, which shows what parameters can be passed to call the portlet as well as examples of calling the portlet from a PL/SQL stored procedure or from a URL.

What Do the Wizards Generate?
Upon completion of the final step of an OracleAS Portal wizard, the OracleAS Portal engine generates PL/SQL code to reflect the choices the developer made on the various pages of the OracleAS Portal wizard. The resulting PL/SQL code is stored in the infrastructure database under the schema the developer selected on the first step of the wizard. Developers often wonder if they can use the OracleAS Portal wizards to generate most of the code needed for the portlet and then go in and customize the resulting code to their specifications. While this certainly is possible, it's highly unlikely that a developer could use this method for two reasons: 1) the generated code is extremely large—even a simple report or form will generate almost 1,000 lines of code and 2) any changes made to the generated code by the
developer will be lost if the wizard is run again. If developers want more control over the portlet than the wizards can provide, those developers would be much more productive using a tool like Oracle JDeveloper to create their portlets.

Creating Your First OracleAS Portal Objects
Earlier in this chapter, in the section titled “Creating an OracleAS Portal User,” we created a user with the ability to create providers. Let’s go ahead and start creating some OracleAS Portal components as that user. Later in the chapter, we’ll take some components and place them on an OracleAS Portal page.

If you are not logged in already, log in as the development user created earlier in this chapter. Select the Navigator link and click the Create New... Database Provider link. Give your Provider a name (no spaces or special characters) and a display name, and select a database schema to use for this application (see Figure 1-22). Click OK.

![Create Portal DB Provider screen](image)

**FIGURE 1-22** The Create Portal DB Provider screen
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

Back in the OracleAS Portal Navigator screen, click the link with your provider’s name. You are taken to a navigator page where you can build the following OracleAS Portal objects:

- **Form**  Forms display a customized form that can be used as an interface for updating tables, executing stored procedures, and generating other customized forms. You can build three types of forms:
  - A form based on a table or view enables end users to insert, update, and delete data in a database table or view.
  - A master-detail form displays a master row and multiple detail rows within a single HTML page. The form contains fields for updating values in two database tables or views.
  - A form based on a procedure enables end users to insert, update, and delete data in a database-stored procedure.

- **Report**  Report objects display data you select from the database table or view in a report. Your report can have a tabular, form, or custom layout.

- **Chart**  Chart objects display data you select from a database table or view as a bar chart. You can also create Java-based image charts.

- **Data component**  Data components display data in spreadsheet format.

- **Calendar**  Calendar objects display data you select from a database table or view as a calendar.

- **Dynamic page**  Dynamic pages display dynamically generated HTML content on a Web page.

- **Hierarchy**  Hierarchies display data you select from a database table or view as a graphical hierarchy of items containing up to three levels.

- **Menu**  Menu objects display an HTML-based menu containing hyperlinked options to other menus, OracleAS Portal database portlets, or URLs.

- **Frame driver**  Frame drivers display a Web page with two frames. End-user queries in one frame control the contents of the other frame.

- **Link**  Link objects display a clickable link that provides a hypertext jump between OracleAS Portal database portlets and other database portlets, database portlet customization forms, or any HTML page.
List of values (LOV)  LOVs enable end users to choose entry field values in a form or database portlet customization form. You can use LOVs when creating database portlets to preselect the possible values in an entry field. The end user clicks the mouse to select a value rather than type it. You can also build LOVs based on other LOVs. LOVs are assigned to fields on a Form or Report and are the only OracleAS Portal components not placed directly on a page.

URL  URL objects display the contents of a URL.

XML Component  XML components display an XML page.

OracleAS Portal Forms
One of the basic OracleAS Portal components is that of an OracleAS Portal Form (Figure 1-23). A form allows end users to interact directly with the database and can be designed to query, insert, update, or delete data, or most likely, to enable the end user to perform a combination of these activities. The OracleAS Portal wizard used to create an OracleAS Portal form, like all of the other wizards discussed in this chapter, can be used to create a component that can be run by itself over the Web and accessed via a Web browser. These components run outside of OracleAS Portal’s security and page structure and can be used to create components that are not placed on OracleAS Portal pages. Running the OracleAS Portal components in this manner is commonly referred to as full page or standalone mode. As this is unusual, the focus of this chapter will be on the creation of OracleAS components that are designed to be used as portlets (i.e., those to be placed on an OracleAS Portal page).

NOTE
To see the steps involved in creating the different types of OracleAS Portal Forms and other objects, go to http://www.tusc.com/. I recommend that you go through all examples referenced in this book in order, as concepts introduced in the early lessons are referenced in the later ones.

OracleAS Portal Reports
The Reports component of OracleAS Portal gives developers the ability to create great-looking OracleAS Portal Reports. Developers can change the look of the report according to values queried (or calculated) from the database, and then they can grant power users the ability to modify the query used to drive the report and can even create links in their reports that allow users to click a report and be taken to another OracleAS Portal component.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

When you select the Create New... Report link in the Navigator, you are presented with a page similar to the one used to create our first OracleAS Portal Form (Figure 1-24).

Your choice of the three different types of OracleAS Portal Reports will depend on your reporting requirements:

- **Query By Example (QBE) Reports**  Choosing this type of report will not only generate the report itself but will also generate a customization screen along with the OracleAS Portal Report. Power users can be granted privileges that allow them to modify what data is returned to the portlet and to insert/update/delete records in the reports' tables.

- **Reports From Query Wizard**  This type of report is similar to a QBE report except for the fact that the customization page contains many fewer...
options—the ability to insert/update/delete records is not available, nor is the ability to change the WHERE clause of the query driving the report.

- **Reports From SQL Query**  This option allows developers to write their own queries that will drive the report.

**NOTE**
To see the steps involved in creating the different types of OracleAS Portal Reports, point your browser to http://www.tusc.com/. It is the author's recommendation that you go through all examples referenced in this book in order as concepts introduced in the early lessons are referenced in the later ones.

---

**FIGURE 1-24**  The first page of the OracleAS Portal Report Wizard
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

OracleAS Portal Charts
OracleAS Portal Charts give developers the ability to graphically display information as a portlet on their OracleAS Portal pages. Like OracleAS Portal Forms and Reports, OracleAS Portal Charts are developed using a series of wizard pages and can be created with various customization options that can then be granted to certain users. As you will see, OracleAS Portal gives developers great freedom in choosing the type of chart to display and what formatting options are to be used when displaying the chart.

To create a new OracleAS Portal Chart, bring up the Navigator by clicking the Navigator link on the top right of any OracleAS Portal page. Click the Providers tab if it is not already selected. Click the Locally Built Providers link, and then the name of the provider you have been using to build the examples so far. On the top left of the page, click the Create New... Chart link.

The first page gives you the option of creating a chart from the Query Wizard or a chart from a SQL Query. Building the chart using the Query Wizard allows you to quickly build and deploy a chart, but developers are restricted to a limited number of features. Building a chart from a SQL Query gives the developer greater flexibility but requires more work (particularly when constructing the driving SQL query). With either selection, you are presented with the now-familiar first page of the Chart Wizard (Figure 1-25) after choosing what type of chart to build.

NOTE
To see the steps involved in creating the different types of OracleAS Portal Charts, point your browser to http://www.tusc.com/.

OracleAS Portal Dynamic Pages
An OracleAS Portal Dynamic Page allows developers to control every aspect of their OracleAS Portal component. Unlike in the OracleAS Portal Reports and Forms wizards, there are no wizard pages here to specify column or page formatting, color of font specification, or portlet settings. While this OracleAS Portal component gives the developer the most flexibility when designing his or her portlet, the developer must handle all details of formatting, validation, and appearance.

OracleAS Portal List of Values
An OracleAS Portal List of Values is a unique component in the sense that, along with OracleAS Portal Links, it is not designed to be placed on a page; rather, the purpose of an LOV is to be used as an attribute to provide data to other OracleAS Portal components. An LOV is invaluable when used in data entry–type applications
OracleAS Portal Dynamic Page
The name Dynamic Page can be a little misleading. Most of the time, developers create these components to be used as portlets that are then placed on OracleAS Portal pages. Placing a Dynamic Page portlet on a page does not affect the characteristics of that page; it's still an OracleAS Portal page. If the Dynamic Page is displayed in standalone mode, then it truly is a dynamic page; otherwise, it is just a dynamic portlet that is placed on an OracleAS Portal page. To see the steps involved in creating the different types of OracleAS Portal Dynamic Pages, point your browser to http://www.tusc.com/.

FIGURE 1-25 The first page of the Chart Wizard
for two reasons: 1) it eliminates incorrect data from being entered, and 2) it eliminates the need to memorize various codes needed to drive applications.

An LOV, after it is created, exists in OracleAS Portal but is not available to be placed on an OracleAS Portal page. Any new OracleAS component can use the LOV, and any existing OracleAS component can be modified to include the new LOV. LOVs can display values in different formats:

- **Combo box** This displays the data in a text box with a drop-down arrow on the text box’s right side. When a user clicks the arrow, the box drops down to display all of the values of the LOV. Only a single value can be selected. Combo boxes are good for application screens that do not have a lot of free space on them.

- **Popup** This displays the data in a text box with a small notepad icon to its right. Clicking the icon opens another window, where the appropriate value can be selected. As in combo boxes, only a single value can be selected. We’ve seen popups for selecting tables or views when we created our example OracleAS Portal Forms and Reports earlier in this chapter.

- **Check box** This displays all of the data for the LOV with a small check box to the left of each entry. Check boxes are good when you want to display all values in the LOV and give the end user the ability to select multiple values.

- **Radio group** This displays all of the data for the LOV with a small circular button to the left of each entry. Only a single value can be selected. Radio groups are good for applications that need to display all LOV data to the user at one time and then allow one selection to be made.

- **Multiple select** This displays the first couple of data elements for the LOV in a text box. The user can scroll through the values and select multiple values by holding down the `SHIFT` key and single-clicking the appropriate values.

Figure 1-26 displays a simple LOV in different formats.

Figure 1-27 shows the two types of LOVs you can build (Dynamic and Static).
NOTE
To see the steps involved in creating the different types of OracleAS Portal LOVs, point your browser to http://www.tusc.com/.

OracleAS Portal XML Components
The first step of the XML Component Wizard (Figure 1-28) shows the ubiquitous naming page of the wizard. Step 2 of the wizard gives developers the opportunity to either enter a URL that points to an existing XML file or place the XML code on the page. As in the dynamic pages created earlier, OracleAS Portal allows developers to
embed SQL or PL/SQL code within the XML text box by surrounding it with the `<oracle>` and `</oracle>` custom tags. You can also make use of bind variables to give your end users another layer of interaction with the portlet. Step 3 lists all of the code between the `<oracle>` and `</oracle>` tags so that you can check it and make any modifications to it.

**NOTE**

To see the steps involved in creating the different types of OracleAS Portal XML Components, point your browser to http://www.tusc.com/.
OracleAS Portal Calendars

A calendar is a graphical object that can be used to display links that reference specific records in your database. By default, the record must have a date field, or a field that can be converted into a date field by way of the TO_DATE SQL function if you intend to use it with an OracleAS Portal Calendar. The calendar then displays with the corresponding links in the calendar, where the end users can drill down for more information. The trickiest part of building a calendar for use in our portal will be constructing the appropriate query to return data that will drive the calendar. This query is similar to the query we constructed when building the OracleAS Portal Chart component earlier. The first step of the Calendar Wizard is shown in Figure 1-29.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

NOTE
To see all the steps involved in creating the different types of OracleAS Portal Calendars, point your browser to http://www.tusc.com/.

OracleAS Portal Hierarchies
A hierarchy is a graphical representation of elements in a list that has been queried from the database. Hierarchies are very useful for things such as employee charts or a list of parts that makes up a large component such as an automobile or a computer server. Hierarchies can also be created with links so that they become a way of allowing end users to interact and “drill down” for more information based on what is displayed in the hierarchy.
The part of creating an OracleAS Portal Hierarchy that can be complex is understanding the fact that the table you wish to base your hierarchy on must have a relationship where values in a table column can be related to those in another column in the same table or another table; for example, relating the values between a primary key and a foreign key. Consider, for a moment, the EMP (employee) table in the PORTAL_DEMO schema that is created in the infrastructure database during the installation of Oracle Application Server 10g:

```
SQL> desc portal_demo.emp;
Name                                      Null?    Type
----------------------------------------- -------- ------------
EMPNO                                     NOT NULL NUMBER(4)
ENAME                                              VARCHAR2(10)
JOB                                                VARCHAR2(9)
MGR                                                NUMBER(4)
HIREDATE                                           DATE
SAL                                                NUMBER(7,2)
COMM                                               NUMBER(7,2)
DEPTNO                                             NUMBER(2)
```

This table contains a recursive relationship between the MGR and EMPNO fields that makes it a good candidate for a hierarchy. Step 1 of the Hierarchy Wizard asks us to specify the name of the hierarchy. Step 3 asks us to specify what table or view the hierarchy will be based on.

**NOTE**

To see the steps involved in creating the different types of OracleAS Portal Hierarchies, point your browser to http://www.tusc.com/.

**OracleAS Portal Menus**

An OracleAS Portal Menu is an HTML page that displays links to submenus, OracleAS Portal database portlets, or external URLs. The menu, its submenus, and any links to OracleAS Portal database portlets or URLs can be secured at the OracleAS Portal role level to prevent access by unauthorized users. OracleAS Portal Menus can display as many as five levels of a menu hierarchy, with each level indented on the menu to the right. Descriptive text can be added to links, and hyperlinks can be added. You can set an overall different look and feel for the menu and its submenus based on a template, or you can set a different look and feel on a submenu-by-submenu basis.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

To create a new menu, click the Navigator icon on the top right of any OracleAS Portal page. Click the Providers tab if it isn’t already selected. Click the Locally Built Providers link and then click the provider you have been using to work thorough the examples. Click the Create New... Menu link on the top left of the page. Menu creation via the wizard only has four steps (the first is shown in Figure 1-30), so it is one of the least complicated OracleAS Portal components we can create.

NOTE
To see the steps involved in creating the different types of OracleAS Portal Menus, point your browser to http://www.tusc.com/.

FIGURE 1-30  The Menu Items and Submenus page of the Oracle AS Portal Menu Wizard
OracleAS Portal URLs

An OracleAS Portal URL is, perhaps, the simplest component you can create in OracleAS Portal. A URL portlet that is placed on a page does not display the HTTP hyperlink; rather, it resolves whatever is in the URL and displays that page in the portlet. To create an Oracle Portal URL, navigate to your provider page. Click the Create New... URL link on the top left of the page.

NOTE
To see the steps involved in creating the different types of OracleAS Portal URLs, point your browser to http://www.tusc.com/.

OracleAS Portal Links

OracleAS Portal Links are a way of tying together your OracleAS Portal components. With links, for example, you can produce a report, as an example, that has one of its columns turned into a set of links. When an end user clicks on one of those links, that user is taken to another component (such as an OracleAS Portal Form) that provides more information about the selected record. There are two aspects of links that make them extremely powerful:

■ **They can be used in multiple places.** Once a link is defined to a particular component, a component’s customization form, or an HTML link, it can be placed on as many components as you like. This is very powerful, as it easily gives you a way to allow a high level of interaction among portlets on your OracleAS Portal and provides a standardized, consistent way of handling this interaction.

■ **They allow parameters to be passed.** Links are “smart” enough to pass the appropriate value(s) to the target (destination) OracleAS Portal component. By doing so, the overhead of complex programming involving parameters is eliminated.

To create a new link, select the Create New... Link link on the Providers page in the OracleAS Portal Navigator to bring up the OracleAS Portal Link Wizard shown in Figure 1-31.

NOTE
To see the steps involved in creating the different types of OracleAS Portal Links, point your browser to http://www.tusc.com/.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

OracleAS Portal Data Components

An OracleAS Portal Data Component is the equivalent of embedding a spreadsheet into your OracleAS Portal pages. The wizard does not prompt you for any source information (such as a database table), since the spreadsheet is not populated upon being displayed on an OracleAS Portal Page. Depending on whether you select a report or a chart in Step 3 of the wizard, you will see an entirely different set of wizard pages for this component.

NOTE
To see the steps involved in creating the different types of OracleAS Portal Data Components, point your browser to http://www.tusc.com/.
When to Use the OracleAS Portal Wizards

The OracleAS Portal Wizards are great tools for developing portlets quickly. In fact, once developers become proficient with these tools, they can create portlets, set the appropriate privileges, and place them on the appropriate page in a matter of minutes. Beginning developers, with just a small amount of JavaScript and PL/SQL knowledge, can create relatively complex Web pages that are secure without the complex knowledge needed for most Web-based applications. No other development tool can provide developers with this type of productivity.

The OracleAS Portal Wizards are also appropriate for prototyping. In today's ever-changing environment, sophisticated users are expecting increasingly complex Web-based applications to perform their job duties. This, combined with ever-shrinking development times, makes the Web-based application developer's job increasingly more difficult. Using the OracleAS Portal Wizards to rapidly prototype Web-based, database-centric applications can greatly reduce the developer's time when it comes to delivering production-quality applications to end users when rigorous end-user requirements cannot be gathered.

Having said that, there are some significant limitations to declarative-based development with the OracleAS Portal Wizards. Any application that requires the developer to have a level of control beyond what is provided in the wizards is not a good candidate. Most developers will hit this limitation relatively quickly. Also, any application that is not strictly database-centric in nature is not a good candidate for declarative development techniques in OracleAS Portal.

OracleAS Portal Page Design

Up until this point, we have looked at creating various OracleAS Portal components. How do we take them and place them on a page to be viewable by our end users? The answer lies in the OracleAS Portal Design Page. To create a new page, enter the Navigator by clicking the Navigator link on the top right of any OracleAS Portal page. Up until this point, all of our OracleAS Portal examples have utilized the Providers tab on this page. To create a new page that can hold content, however, we will select the Page Groups tab (Figure 1-32).

Page Groups

All pages must be associated with a Page Group page. The Page Group tab on the Navigator allows us to create new page groups, where we can then create sub-pages, templates, categories, navigation pages, perspectives, styles, attributes, page
types, and item types. Each of these components will be discussed in depth in Chapters 3 and 4. For now, click the Create New... Page Group link on the top left of the screen. The Create Page Group Wizard is a simple one: It consists of one page that prompts for a name, display name, and language. Only English will appear in the Default Language drop-down box unless you have installed other language packs for OracleAS Portal. Every page group has at least one page associated with it called the root page, and this page is created automatically for you when the page group is created. After entering the necessary information, you are automatically taken to the Page Layout screen for the root page of the group you have just created (Figure 1-33).
The Edit Page Window

You can edit a page in OracleAS Portal one of three ways, depending on the link selected on the top left of the screen:

- **Graphical** This will display a rough estimate of how the page will look as you add regions, content, and portlets to it.

- **Layout** This will display a layout page for adding regions, content, and portlets to a page (developers who have used earlier versions of Oracle Portal will be familiar with this page).

- **List** This provides a convenient way for developers to manage content on a page. It is not much use for managing portlets.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

Click the Graphical link on the top left of the page. By default, there are two regions created for you on the page: a banner region along the top of the page and an empty region below that. The banner region is considered a navigation page, which is one of the subcomponents of a page group listed at the start of this section. After adding some portlets to our page, we will look at editing that part of the page. In the region below the banner, there is a set of nine icons along the top of the empty region. Moving your pointer over each one without clicking it will bring up a tooltip telling you what action clicking each icon will perform. The first (left-most) icon is the Add Item icon. Next to that is the Add Portlet icon.

Adding Portlets
A region, as you will soon discover, can contain items (content) or portlets but not both. If that's the case, why are both the Add Item and Add Portlet icons available for this region? The answer lies in the fact that, by default, the region is created as “undefined,” meaning we can place either items or portlets in the region now. As soon as either an item or portlet is placed in the region, the region type becomes defined as that type. Place a portlet on the region by clicking the Add Portlet icon in the region below the banner.

The Add Portlets page (Figure 1-34) is displayed. From this page, we can select portlets that will be placed in the region we have selected. There are two ways to search for portlets that you would like to place in the region: If you know the name of your portlet (or a part of the name), type that in the search text box at the top of the screen and click Go. You can also search for portlets by looking through the categories provided to you by Oracle (portlets provided by Oracle are called “seeded” portlets):

- Portlet Builders
- Portal Community News
- Portal Content Tools
- Administration Portlets
- Published Portal Content
- Shared Portlets

Or as another option, you can search for portlets you have created by looking in the Portlet Staging Area category.
Clicking the portlet will move it to the right-hand side of the screen. Go to the Portlet Staging Area and click the name of the provider you have been creating your portlets under. You will see a list of the portlets you have created up to this point. Single-click one of them and it will display on the right side of the page. Click OK to return to the Page Edit mode of your page with the portlet displayed graphically (Figure 1-35).

Click the View Page link on the top right of the screen to see how the page will be displayed when end users request it. Click the Back button of your browser to return to the Edit Page window. Click the pencil icon (fourth from the left) in the region where you have just added your portlet. This displays the Edit Region page (Figure 1-36).
Regions

Regions are sections of a page. They can contain either items or portlets, but not both. They can be sized so that they take up a certain number of pixels on a page or a certain percentage of the page. Each region has its own set of attributes that affect how it displays items or portlets contained within it. The Edit Region page allows you to set attributes for a region on the page. The Main tab allows you to define whether the region is titled and how much space on the page it will take up. When we add regions to our pages in the next step, the OracleAS Portal engine will automatically resize existing portlets so that they can fit on the page. The OracleAS Portal engine will evenly divide the page depending on where we add regions, so
we can return to this page to change the region size if we do not want evenly sized regions. The Attributes tab allows you to define what will be displayed, along with which portlets or content that are in the region. When we look at adding items (content) in the next section, we will revisit this page. Click Close to return to the Page Edit page.

Next to the pencil icon for the region there are four icons. These icons are used to add regions to the page. A region will be added in the direction that the arrow is pointing in the four icons. For this example, click the icon with the arrow pointing to the right (the eighth icon from the left). As we mentioned before, the OracleAS Portal engine will automatically size the new and existing regions equally (Figure 1-37).
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

Changing the attributes for one of the regions will resize the others it affects automatically. To see an example of this, click the pencil icon for the leftmost region. Change the Width field on the Main tab to 66%. Click OK to redisplay the page. As you can see, the leftmost region now takes up 66% of the screen and the right-most region takes up 33% of the screen. To verify this, click the pencil icon on the rightmost region and note how it has automatically been resized to have a width of 33%.

**TIP**
Since the rightmost region is an undefined region, the page properties will not display. Click the radio button next to Items and click the Apply button.
We now have a page with two regions on it: a portlet region on the left-hand side that takes up two-thirds of our page and an item region on the right-hand side that takes up one-third of our page. The colors and fonts that are displayed by default are pleasant enough, but we want to make our Portal distinctive. To change the colors and fonts that are displayed, we need to apply a style to the page.

**Styles**

A style is a set of colors and fonts that are used to define the look of a Web page. By default, numerous styles are provided for you when OracleAS Portal is installed. You are not limited to the seeded styles; you can create your own styles and then apply them to pages if you’d like. To create a new style, click the Navigator link on the top right of any OracleAS Portal screen. Click the Page Groups tab and then the link of the page group you have created in this chapter. Click the Style link. You’ll notice that there are no styles there. What about the seeded styles? To see those, you’ll have to navigate up to the root of the Page Groups section by clicking the Page Groups link in the breadcrumb menu. From there, click the Shared Objects link, and then the Styles link. This will display all of the seeded styles available for you. If you choose to create a style here, it will be available to all pages across all page groups. If you choose to create a style under a specific page group, it will only be available for that page group.

**NOTE**

The preceding sentence is true for all objects in the Shared Objects section of the Page Groups tab: Templates, Categories, Navigation Pages, Perspectives, Styles, Attributes, Page Types, and Item Types.

Return to the page group by clicking the Page Groups link in the breadcrumb menu, and then clicking the link of the page group we have been working with. From here, click the Styles link and then the Create New... Style link to create a new style. After defining the name and display name, you are automatically taken to the Properties tab for the new style. Here, you can define virtually every aspect of how things will appear on your page. The first drop-down box on the top left of the screen under the header Style Element Type lists the four types of elements you can affect on the page: Items, Tabs, Portlets, and Common. Depending on what is selected there, the second drop-down box under the header Style Element
Properties will change to reflect the first selection. The elements under the color palette will also change depending on what is selected in the first two drop-down boxes. Table 1-1 lists what can be specified on this page.

Experiment by changing some of the properties to something distinctive (it’s easiest to change something in the Common section such as Background Color) and saving the Style by clicking Close. Return to the root page of the page group by clicking the Page Group link in the breadcrumb menu and then clicking the Edit Root Page link next to the page group you’ve been working with. Click the Style link on the top of the page to be taken to the Style tab for the page. Select the style you’ve just created from the Choose Style drop-down box. Click OK to return to the graphical editing view for the page. Click the View Page link on the top right of the screen (Figure 1-38).

<table>
<thead>
<tr>
<th>Style Element Type</th>
<th>Style Element Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>Group By Banner, Group By Text, Group By Link, Default Attribute, Sub Page Title Associated Functions, Author, Base Item Type, Category, Create Date, Creator, Date Updated, Description, Display Name, Display Name and Image Link, Display Name Link, Document Size, Expiration Period, Expire Date, Help URL, Image or Display Name Link, Item Content, Keywords, Last Updated By, Page, Page Group, Perspectives, Portlet ID, Portlet Name, Provider ID, Provider Name, Publish ID, Score</td>
</tr>
<tr>
<td>Tabs</td>
<td>Active Tab Color, Active Tab Text, Inactive Tab Color, Inactive Tab Text</td>
</tr>
<tr>
<td>Portlet</td>
<td>Portlet Header Color, Portlet Header Text, Portlet Header Link, Portlet Header Style, Portlet Subheader Color, Portlet Subheader Text, Portlet Subheader Link, Portlet Body Color, Portlet Heading1, Portlet Text1, Portlet Heading2, Portlet Text2, Portlet Heading3, Portlet Text3, Portlet Heading4, Portlet Text4</td>
</tr>
<tr>
<td>Common</td>
<td>Background, Region Banner, Region Banner Text</td>
</tr>
</tbody>
</table>

**TABLE 1-1**  
*Element Properties That Can Be Set on the Edit Style Page*
Templates

A template can be used to predefine the tabs and regions on a page. A template can be associated with a page when it is created, or it can be associated afterward. If a template is associated with a page after items and portlets have been placed on it, you will be asked which regions and tabs on the template move the items and portlets. It is much easier to associate a template to a blank page right after it has been created.

NOTE

Pages can also be “detached” from templates.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

The latest version of OracleAS Portal (10.1.4) has two types of templates: Portal Templates and HTML Templates. Portal Templates are created with an OracleAS Portal Wizard. Portal Templates are used to enforce a particular layout, style, set of privileges, and content across multiple pages. As with a standard page, you can divide a Portal Template for pages into regions and tabs, apply a style, grant access privileges, and add items and portlets. Page Designers can also define parameters for the template and use them in conjunction with the portlets placed on the pages that are based on the template. HTML Templates can be created with either an OracleAS Portal wizard or a third-party HTML editor. You can extend an HTML Template using OracleAS Portal substitution tags or your own developed JavaScript. HTML Templates include two subtypes: HTML Page Skins and HTML Content Layouts.

HTML Page Skins can be used to define the appearance of the area surrounding page content. With HTML page skins, you define where the body area of the page is placed. The body area is where portal content displays, that is, the regions, tabs, items, and portlets. You can introduce portal elements into your page skin through the use of HTML page skin substitution tags. Use these to introduce such OracleAS Portal elements as page Edit or Personalize links, the name of the currently logged-in user, the current page’s display name, and so on.

HTML Content Layouts can be used to define a formatting scheme for individual regions. Design HTML templates for either item or portlet regions. Page designers can use HTML to create tables, font designs, colors, and any other encoding or object type you could place in any other HTML template. You could even call a cascading style sheet (CSS) to apply a standard format. HTML Templates are discussed in detail in Chapter 4.

Create a new Portal Template by clicking the name of the page group we have been working with in this chapter, and then clicking on the Portal Templates link. Click the Create New... Portal Template link on the top left of the page. Give the template a meaningful name, display name, and description, and then click the “Make available for use in this page group” check box and click Next. If you would like to restrict the style to maintain a consistent look and feel, de-select the “Enable Pages to Use a Different Style” check box. When you click Finish, you will be taken to a page that looks similar to the Edit Page screen (Figure 1-39).

On this screen you can define regions and tabs, as well as place items and portlets, although the last two actions are rare. Experiment by adding some tabs and regions to this template. Changes are saved automatically as you add regions or tabs to the template. When you’re finished, click the Navigator link in the top right of the screen. Click the name of the page group in the breadcrumb menu and then click the Pages link. Click the Create New... Page link on the top left of the screen. Leave Page Type as standard and enter meaningful information for the name, display...
name, and description fields. Click Next. On the second page of the wizard, select the template you created in the previous step. Automatically, the screen repaints to display what the new subpage will look like (Figure 1-40). Clicking Finish after this step will display the page in graphical edit mode. Note that the developer has no way to delete or add regions (they are “locked” by the template). It is also impossible to add portlets or items to the existing tabs, as they have not been defined as portlet or item regions in the template.

**Navigation Pages**

*Navigation pages* are a special type of subpage that are intended to be used as a navigation area for your portal pages. During creation of a navigation page, the
steps and attributes of the navigation page are exactly the same as a subpage except for the following:

- Navigation pages can only be “standard” pages; subpages can be either “standard” or “URL”-type pages and can have their attributes modified.
- Navigation pages can be created from other navigation pages by selecting another page in the Copy From drop-down box.
- By default, navigation pages are published as portlets with the intent of placing them on a page.
- There is no banner region created automatically.
Let's add a navigation window on the left-hand side of the page. Click the Navigator link on the top right of the page, and then click the name of the page group you have created in this chapter. The page of the Navigator that is displayed lists all of the components that can be created that are associated with a page group. For now, click the Navigation Pages link. By default, there are two navigation pages created for each page group: a banner (which is displayed by default on the root page) and a navbar (which is not displayed by default on the root page). Create a new navigation page by clicking the Create New... Navigation Page link. Enter the name and display name and click Create. Click Close to return to the Navigator, and then click the Edit link next to the navigation page you just created (Figure 1-41).
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

As you can see, this page looks very similar to the edit page screen except that no banner region is created automatically. You can place either items or portlets on the default region and you have full capabilities to create other regions and place tabs on them, but it is far more common to maintain one region and place items (usually links to other OracleAS Portal pages or components) in this region. Click the Add Item icon on the top left of the page, select Page Link from the Content Item Types drop-down box, and click Next. Click the notepad icon next to the Path text box and select an OracleAS Portal page from the list by clicking Return Object next to our selection. Returning to Step 2 of the Add Item Wizard, fill in the fields on this page with meaningful values, and click Finish.

After returning to the edit page screen, click the Navigator link on the top right of the screen. Click Page Groups in the breadcrumb menu of the navigator, then click the Edit Root Page link next to the page group we have been working with in this chapter. Add the navigation page by creating a new region on the left of the screen. Click the Add Portlet icon for that new region and search for the navigation page you just created. Place it in the region by single-clicking it. Click OK to return to the page.

Page Properties and Page Group Properties

Both pages and page groups have properties associated with them that affect how the page looks and behaves. A Page Group setting will affect all pages (root and subpages) associated with that page group. At the top of the Edit Page screen, there are two links: a Properties link next to Page Group: and a Properties link next to Page:. Click the Properties link next to Page Group: (Figure 1-42).

The properties for the page group include the following:

- **Main** This tab can be used to change the page group name, display name, or quotas for the size of items that can be placed on pages in the page group. It can also be used to allow or deny end users the privilege to use or modify page styles.
- **Configure** This tab can be used to allow or deny different page types and different types of content on the page(s) in the page group.
- **Items** This tab allows you to enable/disable item versioning, set characteristics for unpublished (unviewable) items, and purge expired items.
- **Translations** This tab allows you to manage different language packs that may have been installed.
- **Access** This tab allows you to set permissions on the pages in the page group.
Click OK to return to the edit page screen. Click the Properties link next to Page: to go to the properties page (Figure 1-43).

This page shows a slightly different set of properties. Along the top, there are now eight tabs:

- **Main**  This tab allows you to change the name/display name of the page and control its caching behavior.
- **Template**  This tab allows you to attach the page to or detach the page from a template.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

This tab allows you to select a style for the page.

Access  This tab allows you to set access privileges for the page. You can use this tab to override settings made in the Access tab of the Page Group properties described in the preceding list.

Optional  This tab allows you to publish your page as a portlet (which can then be placed on other pages), set default WebDAV properties, enable/disable item versioning, specify page images, and enable/disable links to subpages.
NOTE
WebDAV (the term stands for WWW Distributed Authoring and Versioning) is a standard used to save data to a Web site (as opposed to just reading it). WebDAV is very useful when you have a large amount of content you wish to publish on your site.

■ Parameters  This tab allows you to define parameters for your page. You can then pass those parameters in calls to the page from the various packages in the Portal API. Parameters are most often passed to portlets on the page to synchronize portlet behavior. An example of this might be a stock symbol parameter that can be used to drive quote and news portlets relating to the stock symbol.

TIP
The Portal API is described in the Portal Development Kit (PDK), downloadable from http://portalcenter.oracle.com/.

■ Events  This tab allows you to define events for your page. Events can be used to trigger certain actions when various events happen on the page.

■ Regions  This tab allows you to define a default item and default portlet region for the page.

Exercises
At the end of each chapter, there will be a series of exercises that will explore the topics discussed in the chapter. It is recommended that you do these exercises in order. The first couple of exercises are relatively simple and should take a very short amount of time to perform. The later exercises, which introduce more complex topics, will obviously take longer.

While it is strongly encouraged that you work through the exercises, the author realizes that, depending on the reader’s existing skill level, some exercise topics may be review material. If you wish to skip over some of the beginning exercises and focus on the later ones, you can download the pages and portlets for these beginning exercises from the author’s Web site listed in the Introduction to this book.
Chapter 1: Out-of-the-Box Oracle Application Server Portal Technologies

- Exercise 1: Create a Page Group named “Executive Dashboard.”
- Exercise 2: Create three pages: one that holds only content, named “Content Page,” one that holds only portlets, named “Portlet Page,” and one that holds both, named “Mixed Page.”
- Exercise 3: Place a piece of content on your content page. If you don’t have a common file like a Microsoft Word (.doc) document or an Adobe Portable Document File (.pdf) available, simply create a text file, put some random text in it, save it with a .txt extension, and place that .txt file on the page.
- Exercise 4: Place a portlet on your Portlet Page. If you haven’t created any portlets yet, you can place one of the seeded portlets on the page; seeded portlets are discussed in Chapter 3.

Summary

OracleAS Portal is a feature-rich development environment that gives developers a multitude of tools and constructs that allow the creation and deployment of Web-based applications quickly and easily. This chapter has introduced the basic architecture of OracleAS Portal and the OracleAS Portal components that can be built using the wizards provided by Oracle, but we’ve just scratched the surface of what it is truly capable of. In subsequent chapters, details of OracleAS Portal development, security, and advanced topics will be discussed.