CHAPTER 3

OracleAS Portal Page Design
In the Java development world, there is a common framework for developing Java-based Web applications. This framework is called MVC, which stands for model-view-controller. The thought behind the MVC framework is this: by separating the major parts of an application, that is, the code that interacts with a data source (the **model** part) from the code that handles users interaction (the **view** part) from the code that contains the business logic of the application (the **controller** part), development, modification, and testing of the application is greatly simplified. While there are other development frameworks available, the MVC framework is one of the most popular ones today.

What does this have to do with OracleAS Portal page design? The MVC framework allows large projects to be divided among developers with differing skill sets, all the while using a common development tool (if they so choose). Developers who are familiar with the data source (like an Oracle database) can focus on the model parts of the application, a developer focused on the user interface can focus on the view parts, and developers implementing business rules can focus on the controller parts. Often, a single developer is responsible for all aspects of a Web-based application, but as systems, business rules, industry regulations, and end-user requirements get more complex, this is increasingly rare. For more on the importance of the MVC framework, see the sidebar “The MVC Framework.”

In the same way, OracleAS Portal provides tools within its development environment to support declarative-based development, page design, and content management. By separating the page design from the development of portlets, OracleAS Portal implements its own version of the MVC paradigm: you develop portlets using the OracleAS Portal wizards to model the data source(s) and handle the business logic for the portlet, effectively taking care of the M and C parts of the MVC framework. The page design function of OracleAS Portal handles the V part.

It is important to note that there is no rigid delineation between the model-controller part of portlet development and the view part of OracleAS Portal page design. Portlet developers can, for example, use JavaScript or formatting conditions built into the OracleAS Portal wizards to change how portlets are presented to the end user, which affects the view part of the MVC framework. Page designers can also set up their pages to accept parameters that can then be used to affect what is displayed in a portlet, which affects the controller part of the MVC framework. In Chapters 7 and 8, you will see that when using Oracle JDeveloper to create Java-based portlets, the developer has full control over how the data is presented to the end user. The Portal add-in for Oracle JDeveloper will generate a user interface
Chapter 3: OracleAS Portal Page Design

The MVC Framework
You may be thinking, why is the model part of the MVC framework given such importance? If you are a PL/SQL developer, you know that modeling database structures in your code is as simple as declaring variables using the %TYPE or %ROWTYPE attributes. In object-oriented languages like Java, mapping objects to relational database elements is a lot more complicated. There is also the concept of persistence, defined as the process of having data outlive the object that created it. As Java objects are no longer needed, they are de-allocated from memory; in a complex application, how do you preserve the data you may have manipulated in one of your application’s forms? These are very complex issues for a developer working on a relational database-driven Web application. There are persistence frameworks available to help developers handle the tasks associated with issues like these. Two common ones are Hibernate and Oracle’s offering: Toplink. As part of Oracle’s Java Integrated Development Environment, Oracle JDeveloper, a developer framework called Application Developer Framework (ADF) also handles much of this complexity for you. ADF is discussed in Chapters 7 and 8.

Pages and Page Groups
A page in OracleAS Portal is where developers and content managers place portlets and content for end users to work with. When you type in the URL to get to your portal, an OracleAS Portal page is displayed. After you log in and navigate to places where you can create portlets, add content, administer users, or create pages, you are viewing OracleAS Portal pages.
NOTE
Out of the box, OracleAS Portal contains prebuilt pages and portlets for you to use. These are called “seeded” pages and portlets. There are seeded portlets like a Login portlet and a Search portlet that you can place on any of the pages you create. It is less common to use one of the existing seeded pages in your portal, but you could if you wanted to. A far more common thing to do is to take the structure of an existing page by copying it and modifying it.

As with the OracleAS Portal wizards that are used in declarative development, OracleAS Portal page design is handled through OracleAS Portal. In Chapter 4, we will see how to use a non–OracleAS Portal page design tool such as Dreamweaver to design portal pages, but even in this scenario, we need to use the OracleAS Portal design pages to define the page first.

If you haven’t done so already, log in to your portal. In order to create pages, you need certain privileges. Without those privileges, the options to create and modify pages will not be available to you.

This is a very important concept to note in all aspects of your interactions with OracleAS Portal: If you don’t have the necessary privileges in OracleAS Portal to perform an action, the pages, portlets, and links associated with that action are simply not displayed. You won’t see a message like “You do not have the necessary privilege to perform this action.” This can be frustrating for new users of OracleAS Portal. In the case that the screen you are viewing in your portal does not match the screen shots in this book, most likely it is a permission issue. Talk to your portal administrator or see Chapter 10 for more information.

CAUTION
The portal user has all privileges in OracleAS Portal, so you could log in as that user, but do so with caution. Since the portal user has all privileges, you can do things that will make your portal unusable. For DBAs reading this, logging in to OracleAS Portal as the portal user is like logging in to a database as the sys user. It is also not a good idea to be the portal user when you start development, as complications can arise when it comes time to move parts of your portal from a development environment to a test or production environment. Moving portal pieces from and to different environments is discussed in Chapter 9.
Chapter 3: OracleAS Portal Page Design

If necessary, create a user in OracleAS Portal (or have your Portal administrator do it) and grant that user DBA privileges. After creating that user, log out, and then back in as the user you just created.

In Chapter 1, the concept of the one-to-many relationship was discussed. To quickly review, Oracle uses the one-to-many relationship in many places within OracleAS Portal. These relationships between OracleAS Portal elements provide the following benefits:

- It allows administrators to perform bulk operations on many elements (such as move, delete, or exports) at once.
- It gives administrators the ability to define attributes at the parent level that can cascade down to all child elements associated with that parent.
- It makes the administration of privileges and security much easier. A privilege can be assigned to a parent element and all child elements associated with that parent will automatically inherit those privileges. Individual children can also override those privileges to give administrators an even greater level of security.

The one-to-many relationship that is important here is the one between page groups and pages. In this relationship, the page group is the parent and the pages are the children. Before you can create a page, there must be an existing page group. When a page group is created, a special page is created for that page group automatically. This special page is called the root page. You can create, modify, and delete other pages for this page group, but you cannot delete the root page. For most of the exercises in this book, we will be using a page group called “Dashboard_1.”

Page Groups and Pages
You can think of a page group as a way of organizing the pages in your portal. It is common to set up page groups either by application or by the groups within your organization who will be accessing the pages. For example, you may have a page group called HR for all of the pages that are related to Human Resources in your portal. You might also have a page group called Inventory that has all of the pages with portlets on them that are related to inventory and inventory management. By taking time to organize your portal pages up front, you can save yourself a lot of time later on when it comes time to set up your users and privileges.
There are two ways to get to the page design screen within OracleAS Portal. One method gives you the opportunity to create a new page group; the other does not. After logging in, click the Build tab on the top right of the page. In Figure 3-1, the main part of the screen is devoted to working with existing page groups. Here, you can use the drop-down box to select what page group you wish to work with, and you can create or modify pages, styles, templates, categories, perspectives, item types, page types, or attributes.

**NOTE**
All of the items under the heading Layout & Appearance are discussed later in this chapter. The items under Content Attribution are discussed in Chapter 5.
If this is a new installation of OracleAS Portal, you will see three page groups in the Work In drop-down box: Portal Design-Time Pages, Portlet Repository, and Shared Objects. If this is not a new installation, you will see these page groups plus any more that you or your team have created. The Portal Design-Time Pages are the seeded pages that developers, content managers, and administrators use to build and administer your portal. The page you are currently looking at is one of the Portal Design-Time Pages. The Portlet Repository is a special group that allows you to see a page with a selected portlet on it. From here, you select a portlet and a page is temporarily created with that portlet on it. This is not a common use of OracleAS Portal; most likely, developers will create portlets, create pages, and then place the portlets on a page manually. The final seeded page group is called Shared Objects. When one of the categories in Figure 3-1 (styles, templates, types, etc.) is created in a page group, it is only available to pages within that page group. Anything created in the Shared Objects page group becomes accessible to all page groups.

**TIP**

The portlet repository page group is there to allow developers and page designers to organize the portlets in any way they choose. Changing the page hierarchy and/or moving portlets around will change the portlet selection window when adding portlets to pages. This allows developers and designers to tailor the portlet selection experience in any way they want. The fact that you can preview the portlet directly from the repository page is an additional benefit.

Since we are unable to create page groups here, it is time to explore the second way of getting to our design pages (users with the right privileges can create a page group directly on the builder page). Click on the Navigator link in the top right of the portal page. In Figure 3-2, you’ll see the Portal Navigator. From the Portal Navigator you can create Page Groups, Pages, all of the items associated with pages (Styles, Items, etc.), Providers (which, in the one-to-many relationships we’ve seen so far, are the parents to individual portlets), Portlets, and Provider Groups. You can also query, create, and modify database objects in the infrastructure database (provided that you have the necessary permissions, of course). Since the Portal Navigator is so flexible, most developers, page designers, and content managers do most of their work here, although you can certainly use the Portal Builder if you are more comfortable with that. You’ll notice that there are no administration functions (create users/groups, etc.) here, so administrators will do most of their work in the Portal Builder.
Click on the Page Groups tab if it’s not already selected. Here, you’ll see a list that corresponds to the drop-down box in Figure 3-1. Click the Create New Page Group link in the top right. You are taken to a simple wizard that asks you for the name of your page group. As mentioned earlier, creation of a page group will also automatically create a root page associated with this page group. The Default Language drop-down box will contain only English unless you have installed other language packs. Type **Dashboard_1** in the Display Name field and click Create. You will see a screen similar to Figure 3-2.

What just happened? After you specified the name of the page group, the page group itself was created, the root page for the page group was created, and you were taken to the Edit page for the root page. Click the Navigator link in the top right—you will see a new page group called “Dashboard_1.”

![Dashboard_1 root page](image)

**FIGURE 3-2** The Edit page for the Dashboard_1 root page in Graphical mode
Chapter 3: OracleAS Portal Page Design

NOTE
You may be wondering where all of these things are created. All of it is created and stored in the infrastructure database of your Oracle Application Server installation. The infrastructure database holds three basic pieces of information: 1) security information from the LDAP server installed with Oracle Application Server, 2) clustering information for the instances of the Oracle Application Server, and 3) Portal information—content, templates, page design, etc.

Clicking the Dashboard_1 link shows all of the OracleAS Portal objects that are associated with that page group:

- **Pages** This link lists all of the subpages associated with this page group. Click that link now. You’ll notice that no pages are displayed. As an unusual quirk of OracleAS Portal, you cannot edit the root page from here. The only way to edit the root page is to go up one level (you can do this by clicking the Page Groups link in the top left of the page in the breadcrumb menu) and then click the Edit Root Page link in the center of the page next to the page group you are interested in.

- **Portal Template** This link allows you to create a template that can be used to create other pages quickly and easily. For example, you can create a portal template that has your company logo along the top of the page and a legal disclaimer at the bottom. You can then use that page template to create other pages without having to go through the effort of defining the logo and legal information over and over again.

- **HTML Templates** This is one of the most exciting new features of OracleAS Portal 10.1.4. By using HTML Templates, you can use a page design tool like Dreamweaver to create your pages and use them in OracleAS Portal with minimal modification. There are two types of HTML Templates: HTML Page Skins, which are used to add a “wrapper” of HTML around a page, and HTML Content Layouts, which are used to format a region. HTML Content Layouts are named a little misleadingly, as they can be used for regions that contain content and regions that contain portlets.
NOTE
Regions will be discussed shortly, but for now, understand that regions are parts of a page that can be defined to hold either content or portlets, but not both. To revisit the one-to-many relationship, regions are the children and pages are the parents.

- **Categories** As your portal grows, you may have large amounts of content. Categories are used to help classify content that users can search through. A particular piece of content (referred to, in Oracle's documentation, as an "item") can only belong to one category.

- **Navigation Pages** A navigation page is a special type of page that can be added to other pages to provide a consistent set of navigational elements. These pages differ from other pages in that they are excluded from searches and bulk actions performed on pages in the page group. Although you can add content or portlets to a navigation page, the idea is to add navigation pages to other pages.

- **Perspectives** Perspectives are very similar to categories in the sense that they help organize the content in your portal. The key difference between the two is that an item can belong to multiple perspectives. Oracle separates these two classifications by saying that categories help organize items according to their content, while perspectives organize items according to who might be interested in them. For instance, you may have a technical note for one of your software products. You can set up a perspective for the product and another one for the operating system it runs on. The technical document could then be associated with two perspectives: "Product x version 2.0" and "Red Hat Enterprise Server 4.x." End users can then use perspective filters in the Advanced Search portlet (a seeded portlet available for you to place on your pages) to narrow their search automatically to the perspectives they are interested in.

- **Styles** Styles define visual attributes of your pages and the things placed on them such as portlets, items, tabs, etc. Each one of the things you can place on a page has further attributes you can define for your style. As an example, for portlets, you can define the portlet header color; the color, font, font size, and font style of the text in the portlet header; the subheader colors and text attributes; etc. If the developer of the portlet has defined
things like these and the page designer has also defined a style for the page, the page style will override the developer’s settings.

**Attributes**  
An attribute is an element that takes a value and is associated with an object, such as an item, a region, a page. An example of such an element is Author, whose value is typically the name of the object creator. Typically, an attribute value is provided by a user, though there are some attributes that provide their own default values. For example, the Publish Date attribute has a default value of the current date and time. In most cases, users can revise an attribute’s default value. There are two types of attributes, with all attributes falling into either one or both types: Content attributes are associated with item types and page types, and store information about an item or page, such as the associated category, description, or perspectives. These attributes are included in the add and edit screens where users can provide information about the item or page they are adding or editing. Page group administrators can create their own item types and page types and specify exactly 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. Display attributes 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. Note that some content attributes, such as author and description, are also display attributes. Any custom attributes created by the page group administrator are also display attributes.

**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. Base page types are the page types included with OracleAS Portal. Following are the five base page types:

- **Standard** Displays items and portlets.
- **URL** Displays the contents of a particular URL.
- **Mobile** Displays items 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 JavaServer Page (JSP).
TIP

If you want to store more information than the default page types allow, you can create a custom page type. You base a custom page type on one of the base page types. The custom page type automatically inherits all of the base page type’s attributes. After creation of a custom page type, developers can edit it to add attributes that are specific to your requirements.

- **Item Types**  
  Item types define the content of an item and the information that is stored about an item. Items in OracleAS Portal are based on item types. Items are one of the basic components of a portal page. The information stored about an item is determined by the attributes associated with the item type.

- **Base Item Types**  
  Base item types are the item types included with OracleAS Portal. There are two types of base item types. Content item types allow users to add content (for example, images, documents, or text) to a page. Navigation item types allow users to add navigational elements (for example, a login/logout link, basic search box, or a list of objects) to a page. Base content item types include:

  - **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.

  - **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.

  - **Base Item Link**  
    Links to another content item (file, text, URL, image, and the like) within a page group and displays the content of that item or a link to the content, depending on the source’s display option.
Base Item Placeholder  Identifies where the content from items that use a Portal Template display in relation to the rest of the template content. Select default content for the Item Placeholder; choose from file items of type text/html or text/plain, text items, PL/SQL items, and URL items.

Base navigation item types include the following:

- **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 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.

The base content item types are not actually available for users to add to pages. Instead OracleAS Portal provides extended item types based on the base content items. These include:

- **File and Simple File**

- **Simple Image**
If these extended item types do not provide enough flexibility, you can further extend most of them to meet your requirements, provided you have the appropriate privileges. To extend these item types, you must have at least the page group privileges Manage Classifications and View on the Shared Objects page group. You can add different attributes to the item types to store exactly the information that you want. Developers and designers can also add calls to PL/SQL and HTTP procedures and even pass attributes to the parameters of those procedures.

Edit the root page of the “Dashboard_1” page group by clicking the Page Groups link in the top-left corner, and then clicking the Edit Root Page link to the right of the Dashboard_1 link. Along the top left of the page, there are three links next to the Editing Views: label: Graphical, Layout, and List. Clicking each one of these links gives you a different way of editing your page. Each of these editing views has its advantages and drawbacks:

- **Graphical** As you place portlets and content on the page in this view, they will display along with some extra editing icons and links. The advantage of this view is that you can see a rough estimate of what your page will look like to end users while you are editing it. The drawback of this mode is that you cannot do bulk actions such as delete or move a bunch of portlets and content with one action. Also, for pages with many portlets and much content, the amount of information on the screen can become overwhelming.
**Layout**  Layout mode does not display the portlets or content, but rather a one-line title of said portlet or content (Figure 3-3). This view can be very useful for bulk operations such as moving or deleting content and portlets and can make the design of complicated pages easier.

**List**  List mode is great for content (Figure 3-4). You can move, delete, and expire content easily as well as get more detailed information on the content such as author or date added. List mode has no functionality to administer portlets.

---

**FIGURE 3-3**  The root page in Layout mode
Page and Page Group Properties

Click the Graphical link on the top right of the page if you haven’t already done so. Just below the Editing Views line on the top right is a line with five links: Page Group Properties, Page Properties, Style, Access, and Create Sub-Pages. The following sections describe the first two of these at length.

Page Group Properties

This link will take you to a tabbed screen where you can define various properties for the page (Figure 3-5). The tabs include

- **Main** This tab gives you the opportunity to change the name of the page, the display name (what displays in the Portal Navigator), the amount of content
that can be added to the page group, whether users can change page styles, and whether tab persistence (whether OracleAS Portal “remembers” what tab was last selected when you re-visit a page) is active.

- **Configure**  This tab gives page designers the ability to set various properties for the page group:

  - **Page Defaults**  Displays options for defining a default style (styles are discussed at the end of the chapter), defining a default navigation page, and defining a default template for all pages in the page group.

  - **Types and Classification**  Displays options for defining the types of pages permitted in the page group, the types of items that can be used when adding content to any of the pages in the page group, and the categories and perspectives that an item can be assigned to.

  - **Edit Mode**  Displays options for what edit modes are available, what the default edit mode for a page should be, and the number of items and attribute columns of those items that should be displayed in list mode.

  - **View Mode**  Displays the option of whether to define a default item that toggles between showing and not showing content on a page. See the sidebar “More about View Mode.”

  - **Parameters and Events**  Displays the option to toggle between allowing and restricting parameters and events for all pages in the page group.

  - **Content Management Event Framework**  Enable this option to allow the integration of external processes into the portal.

---

**More about View Mode**

View mode allows developers or designers to treat the page group more like a file system if they want to use OracleAS Portal as a deployment platform for static HTML files. This is similar to the way static Web sites work in that if the OracleAS Portal Page Engine finds an index.html file in a folder, it uses that as the page to display if the URL only goes down “into” the folder. As a developer or designer, you can enable View mode and then use “index.html” as the default item—a URL to the page will show the contents of “index.html” rather than the page itself. This is a side effect of using the page as both a container (or folder) and a display vehicle.
Approvals and Notifications  Use this to enable or disable item approvals and item notifications in the page group. When approvals/notifications are enabled, a new tab called Approval displays in the Edit Page Group tab set, on which you can define the approval process and specify notification recipients for the page group.

JSP Access  This option specifies whether JSPs can access this page group.

URL Rewrite Rules  This option specifies the rewrite rules for the path-based URLs in this page group.

Items  This tab gives you the ability to specify item versioning, whether unpublished items are retained or displayed, how long the New icon will appear next to new content, whether to purge old content, and whether to enable the rich text editor when adding text items.

Translations  This tab is only meaningful if you have multiple languages installed. You can specify what languages you would like displayed and how the page translates to that language. Languages other than English are installed by using the Oracle Application Server installation program.

Access  This tab allows you to define who has access to a page and what type of access that user has. This is a confusing concept for many beginning OracleAS Portal developers and administrators. The access privileges at this level grant access only to the page, not necessarily to the content or portlets on the page. As an example, I can grant View privilege to user Kelsey on the page, but a portlet placed on that page may not have given user Kelsey the privilege to see it. When trying to view the page, user Kelsey sees a blank page. She has privileges on the page, but not on the portlet that is placed on the page. While the transparency of OracleAS Portal’s security mechanism is a blessing to developers and administrators, it can be frustrating to new users accustomed to No access granted error messages. Security is discussed in Chapter 10.

Approval  This tab is only visible if Approval and Notifications has been selected in the Items tab. This tab allows administrators to set up a list of users (or groups of users) who need to review and approve a piece of content before it becomes available on the OracleAS Portal page. Approvals are discussed in Chapter 5.

Page Properties
These properties, which have many similarities to the Page Group properties, apply only to a specific page (Figure 3-6). Where Page properties duplicate Page Group
properties, properties set at the Page level will override those set at the Page Group level.

- **Main**  This tab displays options that can be used to change the page name, change the caching rules, or set a time-out for page assembly. For more information about caching, see the “What Is Caching?” sidebar.

- **Template**  This tab provides the option to use a template for the page. This is an example of the potential for Page properties to override Page Group properties. If a template is specified at the Page Group level and is not specified here, this page will inherit the template from the Page Group. If a template is specified at the Page Group level and a different one is specified here, the one specified here will override the Page Group template setting.
What Is Caching?
Caching is a complex subject, but for now, understand that caching is a way to speed up OracleAS Portal by “saving” previous versions of pages so that when they are requested after the first time, the OracleAS Portal engine can display the “saved” version without having to reconstruct all of the pieces that make up the portal page. There is also a component of the Oracle Application Server called, surprisingly enough, Web Cache that separately helps this process. To further complicate matters, most browsers (like Microsoft’s Internet Explorer and Mozilla’s Firefox) have caching mechanisms built into them also. Caching, performance tuning, and monitoring of OracleAS Portal are discussed in Chapter 9.

- **Style**  This tab provides the same functionality as the Template tab, except in relation to styles.
- **Access**  Like the Template and Style tabs, this tab allows you to define access privileges for the page. Again, it can be used to inherit or override settings from the Page Group.
- **Items**  This tab allows you to set item versioning options, the ability to define a default template for items, and whether to define a default region on the page to hold items and WebDAV options.

**NOTE**
WebDAV stands for Web-Based Distributed Authoring and Versioning. It is a set of extensions to the Hypertext Transfer Protocol (HTTP) that allows users to collaboratively edit and manage files on remote Web servers. By adhering to this standard, OracleAS Portal allows any program that is “WebDAV-aware” to communicate with OracleAS Portal. Oracle provides a WebDAV tool called Oracle Drive that makes the transfer of files to OracleAS Portal repositories simple. Oracle Drive is discussed in Chapter 5.

- **Optional**  The attributes defined on this page allow you to define the entire page as a portlet (which can then be placed on other pages) and to set keywords, categories, and perspectives (which can help users search through large portal sites). You can also define an image to represent the page that is used in certain portlets, listing all of the objects in a portal and whether to display subpages of a page.
Chapter 3: OracleAS Portal Page Design

- **Parameters**  Page parameters are used to pass values to a page. Parameters can be used to change information displayed in accordance with where the page is called from or who is viewing the page.

- **Events**  An event is a user action defined by a portlet developer. User actions include clicking a link, a button, or some other control on a Web page. A page designer can specify that an event forces the reloading of the current page or the loading of another page, optionally passing parameters to the newly loaded page. A portlet’s events are specified in the provider.xml file (see the sidebar “The provider.xml File”).

**The Remaining Links**
The remaining three of the five links mentioned on the Dashboard_1 root page design screen are as follows:

- **Page: Style**  Takes you to the Style tab on the Page Properties page.

- **Page: Access**  Takes you to the Access tab on the Page Properties page.

- **Create: Sub-Pages**  Takes you to a wizard where you can create a sub-page (a page that inherits its attributes from the root page and Page Group).

**The provider.xml File**
Chapter 1 described Oracle’s one-to-many philosophy when designing OracleAS Portal. In the context of portlets, every portlet must be associated with a provider. In this scenario, the provider is the parent (the “one” in the one-to-many relationship) and the portlets are the children. When you are ready to create a declarative portlet using the OracleAS Portal wizards, the process of creating the provider handles the creation of all of the necessary configuration files (of which the provider.xml file described here is one) behind the scenes. The provider.xml file is used to define how the provider will function. When creating Java-based portlets, you must explicitly declare the provider.xml file before deploying your application (deploying is a fancy word for moving all of the files that make up your application to the proper directories of the application server). In Chapter 7, the structure of developing Java-based portlets will be discussed and we will see how the Portal Add-in Wizard for Oracle JDeveloper handles many of the implementation details for you.
Page Design

After all of the attributes for your Page Groups and Pages have been set, it’s time to start designing your OracleAS Portal page. Earlier in Figure 3-2, a new root page is displayed. There are standard elements available when editing a page in graphical mode to assist your development. Along the top of the page, there is a header that starts with “Edit Page Dashboard_1 root page.” Below that, there is a rounded horizontal box with Editing Views in the top left, which contains links discussed in the preceding section of this chapter (Figure 3-7). These elements are not part of your final page. They are only visible when editing the page. To see what the page will look like to end users, click the View Page link in the top right of the page at any time. Clicking that link will show you a blank page (since we haven’t added anything to it yet).
Below the horizontal box is a section of the page with eleven small icons on the top left, a graphic that says “Oracle Application Server Portal,” a title in the middle that says “Dashboard_1 root page,” and six links on the right-hand side (Figure 3-8). What is this and where did it come from?

It was mentioned earlier that a page cannot exist without a page group. It was also stated that all page groups must have at least one page associated with them, the root page. So when the page group is created, one default page is created automatically. In addition to this, another page element is created: a default navigation page. The default navigation page is given the name of <Page Group Name> banner, so in this case, if we go to the Portal Navigator and click the Dashboard_1 link, and then Navigation Pages, we’ll see an entry entitled “Dashboard_1 banner.” Navigation pages aren’t really pages, per se—their intention is to be placed on a page. Clicking the Edit link in the middle of the page will display a graphical editing page for the navigation page that is similar to the root page we have been working with (Figure 3-9).
After viewing this navigation page, return to editing the root page for Dashboard_1. If you don’t want to use the banner, there are several ways to dispense with it:

- You can select Page Group Properties, then the Configure tab, then the Edit link under page defaults and change “Default Navigation page” to None. This will stop the banner from being placed on all new pages created in that page group.

- You can, on that same properties page, use the radio button to select either an HTML Page Skin template or Portal Template in the Default Template section of that page. This will base all new pages for that page group on the template selected. Portal and HTML templates are discussed in Chapter 4.

- You can delete the banner by clicking the small red X in the top left of the region. The banner still exists and is still part of the page group—it just isn’t on the page any longer.

**CAUTION**

*If you delete the banner, you won’t be able to edit the page directly any more. You can only edit it via the OracleAS Portal Navigator.*
If you like the banner but want to change it (let's say, by replacing the Oracle Application Server Portal picture with your company logo), you can edit the navigation page. Any edits you make there will automatically be reflected on all of the pages to which that Navigation page has been added.

Under the banner section, there is another horizontal section with only nine icons in its top left and text that says, “This region is empty. You can add content to this region by clicking the Add Item or Add Portlet icon above.” Now is a good time to discuss regions.

**Region Properties**

Regions are a way to divide your page into different sections. A region can divide the page horizontally or vertically, but not diagonally. You can have multiple regions on a page, and each of those regions can be defined to contain either portlets or content, but not both. Regions, like pages and page groups, have their own properties. By default, all new pages contain two regions: a region containing the banner (a navigation page, discussed in the previous section) and an empty, undefined region below it, unless you have specified an HTML Template or Portal Template in the Page Group properties.

The four icons to the left of the red X in the top left of the region (shown in the following illustration) give page designers the ability to create new regions on the page.

Clicking the icon just to the left of the X will create a new region to the right of the existing region. The OracleAS Portal Page engine will evenly divide the page so that the existing region will now occupy 50 percent of the page and the new region will occupy 50 percent of the page. Clicking the same icon (the one with the arrow pointing to the right) in either region will create a third region. All regions now occupy 33 percent of the page (Figure 3-10).

In the left-most region, click the icon with the arrow pointing down. A new region is created below with the same screen dimensions as the region above it. In effect, by dividing the first region three times, three columns were created on the page. To see the effect of this, click the icon with the arrow pointing to the right of the region in the second row. A new region is created to the right, but instead of taking up 50 percent of the entire screen, it takes up 50 percent of the column that takes up 33 percent of the entire screen (Figure 3-11).
Remove all but one of the regions by clicking the red X in all of the other regions. Note how the regions automatically resize themselves as the others are deleted.

FIGURE 3-10 The result of creating two new regions to the right of the existing region

FIGURE 3-11 Dividing regions
We now know what the five right-most icons for each region do. Starting from the left, the icons provide the following functionality, as shown here:

- For an empty, undefined region, the icon that looks like a sheet of paper with a plus sign in front of it and the icon that looks like two small boxes with a plus sign in front of it will be visible. The first icon allows you to add an item (content); the second allows you to add a portlet. Once an item or portlet is added to a region, the region type changes to reflect the addition. Once the region is defined as a type, only the appropriate icon shows up in the graphical editing screen for the page.

- The icon that looks like a folder with a plus sign allows you to add a tab to the region. Tabs can be used to further subdivide a region. Like regions, each tab can hold items or portlets but not both, so it is possible to create a region with one tab for content and another for portlets. Tabs can hold other tabs and other regions and also have their own properties.

**NOTE**

One of the great challenges of working with OracleAS Portal is finding where the necessary changes need to be made in order to get your portal to do the things you want it to do. Just in terms of properties, we’ve seen how there are Page Group Properties, Page Properties, and Region Properties. As you work more with OracleAS Portal, finding the correct place to make modifications will become second nature.

- The icon that looks like a pencil allows designers to view and edit the properties for the region. Just like Pages and Page Groups, Regions have their own set of properties. A region has many fewer properties than a Page or Page Group, however.

**Region Types**

When a new region is created, it is created as an Undefined type. As mentioned earlier, a region can contain portlets or items, but not both. As soon as either an item or a portlet is placed in a region, it becomes a region of that type. You can also define a region to be of a certain type before placing anything in it by clicking the pencil icon in the region to edit its properties and manually specifying what type of region you would like it to be.
There is also a fourth type of region: a subpage links region. This is a special type of region designed to hold links to the subpages of a page. If a region is defined as this type, it will automatically be populated with links to all subpages from the page you are working on (as a designer) or viewing (as an end user).

Once a region has been defined as a particular type, designers no longer have the ability to place a different type of OracleAS Portal element in that region. To see how this works, click the Add Item icon in one of the regions on your page. Select Content Item Types and File on the first page of the Add Item Wizard. Click Next. On the Item Attributes page, click the Browse button next to the File Name box and select a file from your hard drive. Since this is just a demonstration, pick a small text file; if you don’t have one available, use an editor like Notepad to create a small text file. Leave all of the other prompts the way they are and click Finish. OracleAS Portal will now copy this file from your hard drive and transfer it up to the OracleAS Portal repository. It also returns you to the page you were working on (Figure 3-12) showing a link for the text file you’ve added. The important thing to note here is how the region is now defined as an Item region and the icon for adding portlets is no longer available for that region.

TIP
Don’t worry if the screens to add items seemed complicated. Adding content is discussed in detail in Chapter 5.

FIGURE 3-12 The Page Design screen with an Item added to a region
Adding Portlets to Your Pages

Portlets are reusable pieces of code that can be placed on a page. Out of the box, OracleAS Portal contains portlets that developers and designers can use. These portlets, called seeded portlets, can be used just like any portlets that developers at your site create. To add a portlet, find a region that is defined as either a portlet or an undefined region. If you don’t have a region like that on your page, create a new region by clicking one of the four icons that create a new region on your page. After clicking the Add Portlet icon, you will see a page similar to Figure 3-13.

This page allows you to select a portlet from the Portlet Repository to add to your page. The Portlet Repository is where portlets are stored and organized. Out of
the box, OracleAS Portal provides numerous seeded portlets organized into six categories:

- **Portlet Builders**  All of the screens we have been viewing (the Edit Page screen, the Add Portlet screen, the Add Item screen, etc.) are themselves OracleAS Portal pages. All of the portlets on these pages are stored here in the Portlet Builders category. Normally, a portal developer would not place a Create Portal Form or a Create Portal Report portlet on an end-user page (although you could, if you really wanted to), so this category may seem as if it would not be used much. There are, however, some very important seeded portlets in this category of interest to OracleAS Portal developers and page designers:

  - **OmniPortlet**  As mentioned in Chapter 1, Oracle is moving away from the declarative-based OracleAS Portal wizards and encouraging developers to use tools like Oracle JDeveloper to create portlets. If you still prefer declarative-based development, Oracle provides a tool called OmniPortlet that gives developers access to some advanced OracleAS Portal features while still providing a declarative development environment. It is by selecting this link that developers get access to this functionality. OmniPortlet is discussed in detail in Chapter 6.

  **NOTE**  Although I keep hammering away at the fact that Oracle is moving away from declarative-based development, it is important to note that the declarative wizards are still fully functional with OracleAS Portal release 10.x. This book does not discuss those wizards beyond what was covered in Chapter 1, however.

  - **Web Clipping**  This portlet allows developers and designers to clip (copy) a Web page (or, what is more important, part of a Web page) and place it on the OracleAS Portal page they are designing. The clipped page can be an internal OracleAS Portal page, an internal non-OracleAS Portal page, or an external Web page. The Web Clipping portlet is a great way to assemble an OracleAS Portal page from existing sources without any programming skill needed.

  - **Survey Builder**  This folder contains four seeded portlets for creating and reporting on surveys you can build into your portal. Many Web sites contain either a human-interest-type question or a question related to the relevance
of the site end users are viewing. These portlets allow you to develop a survey and run reports against those surveys.

- **Portal Community News** The portlets in this category pull information from Oracle’s Portal site, http://portalcenter.oracle.com/. When you place these portlets on a page, various pieces of news and announcements are queried from Oracle’s Web pages and displayed on your page.

- **Portal Content Tools** The portlets in this category perform various features related to content and content management. Some of the most useful are:
  
  - **The Search portlets (Advanced Search, Basic Search, Custom Search, and Saved Searches)** You can use these portlets to give your end users the ability to search through content that has been added to your site. The search capabilities of OracleAS Portal become invaluable as content is added to your site. Searching content will be discussed in detail in Chapter 5, but for now, know that the search capabilities of these portlets include the ability to search within Microsoft Word documents, Excel spreadsheets, PowerPoint presentations, and Adobe PDF Documents, among other types. There are also classifications (categories and perspectives) that give content managers further abilities to segment content and allow users to narrow their searches.

  - **Approvals (My Approval Status, My Notifications, and Pending Approvals Monitor)** These portlets play an important role in the approval of content to be displayed on a page. The page administrator can set up an approval chain which requires approval from users (or groups of users) within OracleAS Portal before a piece of content can be displayed to end users. These portlets give page designers a way of quickly constructing a page so that the users know when and if they have been assigned a piece of content to review and if an item has or has not been reviewed by those necessary for approval.

- **Administration Portlets** These portlets are used to perform administration on various parts of your portal. All of these portlets are already contained on pages that only users who are members of the portal_administrators group are allowed to view. It is unlikely that a developer or page designer would place any of these portlets on a page except where noted here. Administration portlets are divided into four subcategories:

  - **SSO/OID** This group contains the security portlets for OracleAS Portal. Under most circumstances, none of these would need to be placed on a page, other than the Login portlet, which might be placed on the first page end users would see (commonly referred to as the “landing page”). You don’t have to use the seeded login portlet if you don’t want to—it is fairly
common for developers to write their own. SSO (Single Sign-On) provides a mechanism for users to only have to sign in once and access data across heterogeneous systems. OID (Oracle Internet Directory) is Oracle’s name for their implementation of the Lightweight Directory Access Protocol (LDAP) built into Oracle Application Server. OID and LDAP are discussed in Chapter 10.

- **Portal** This group contains portlets related to internal OracleAS Portal activities such as listing database providers or clearing the Portal page engine cache.

- **Database** This group contains portlets that interact with the infrastructure database that is installed automatically with Oracle Application Server 10g. There are portlets to create and change schemas, list database information, and show database storage statistics. Depending on the privileges of the OracleAS Portal user, it may be advantageous to allow direct interaction with the infrastructure database, but this functionality comes with a stern warning: by giving users the ability to modify data in the infrastructure database, you risk the possibility of data being modified that makes your OracleAS Portal (or worse, your entire application server) unusable.

- **Oracle Reports Security** This group contains portlets that allow you to set up rules for how Oracle Reports can be run within your portal. Note that this does not relate to the OracleAS Portal Reports Wizard briefly discussed in Chapter 1. Rather, these portlets relate to the integration of reports developed with the Oracle Reports product that is part of the Oracle Developer Suite. Integration with common Oracle development tools such as Oracle Forms, Oracle Reports, and Oracle Discoverer is discussed in Chapter 11.

- **Published Portal Content** One of the most interesting things you can do within OracleAS Portal is to check the Publish As Portlet check box in the Optional tab under the properties for a page. Checking this box allows you to treat the entire page as a portlet that can now be placed (as a portlet) onto another page. Any pages that have the Publish As Portlet check box checked will show up in this group. Navigation pages (like the banner at the top of the page that is created by default) have the Publish As Portlet check box selected by default, so they also show up in this category.

- **Shared Portlets** If appropriate, OracleAS Portal Portlets can be customized by users to their individual needs. Developers can allow or deny certain aspects of the portlet to be modified. Administrators can then allow or deny users or groups of users the ability to see and modify those customizations. The customizations are stored in the security mechanism for OracleAS...
Portal, so when a user makes a customization and then logs in to OracleAS Portal from a different computer, the customizations are preserved. An example of a customization would be the ability to change the WHERE clause on a report. Users may want to customize a report to only show information for their department. These customizations can also be saved as shared portlets for other users. Any customized portlet that is saved will show up in this group.

The seventh category, Portlet Staging Area, is where portlets that you create will be stored. The portlets there are grouped by provider. If you don’t know which provider the portlet you wish to add to a page is stored under, or you just don’t feel like searching through the repository for the portlet you’re looking for, you can always use the search box at the top of the screen.

**NOTE**

Providers, discussed briefly in Chapter 1 and in more detail in Chapter 7, are a way of grouping portlets together. Like all of the one-to-many relationships in OracleAS Portal (the provider being the “one” and the portlets being the “many”), any attribute defined at the provider level cascades down to the individual portlets, unless explicitly overridden at the portlet level. This mechanism makes the setting of privileges much easier.

When you click the portlet you wish to add, it appears on the left side of the screen. You have basic control as to where the portlet is placed in the region by using the up and down arrows on the right side of the Selected Portlets box, but that’s about it—the OracleAS Portal page engine will do the work of arranging your portlets in the region. The red X on the right of the Selected Portlets box removes a portlet from the page. Select two or three of the seeded portlets and click OK.

**TIP**

It’s important to remember that although the red X on the Add Portlets To Region page removes the portlet from the page, it is still there in the Portlet Repository.

You are returned to the Page Edit screen. In Figure 3-14, I have added the Login and Advanced Search portlets to my region. Since I’m already logged in, the Login
portlet displays only a Logout link. If I were to make this page available to the “Public” user (i.e., a user that has not logged in to OracleAS Portal) and displayed this page before logging in, a prompt for username and password would be displayed.

Clicking the Layout link at the top left of the page brings us to the Layout display of the page (Figure 3-15). While there is certainly less information and no indication of how the final page will be displayed to end users, this page is useful when the page you are designing becomes so complicated that it becomes impossible to make sense out of it anymore. This view is also useful if the need should arise to move a portlet (or set of portlets) from one region to another. As you can see from the icons above each region, the functionality of adding and removing portlets and regions is the same here as it was when editing the page in graphical mode.
Other Page Elements

If you click the Navigator button in the top right of any page, then the Page Groups tab, and then the page group name (for the examples in this book, we’ve been using the “Dashboard_1” page group), you will be taken to a page that allows you to define other page elements that are associated with your page group:

- **Pages**  This allows you to create subpages off the root page (discussed in Chapter 4).

- **Portal Templates**  A template allows you to define the look and feel of a page. Subsequent pages can be created on the basis of this template, saving the page designer a great deal of work. Another powerful feature is that the application of templates to a page is dynamic, so that if a change is made to the template, all pages based on that template are immediately updated. Portal Templates are discussed in Chapter 4.
■ **HTML Templates**  The Portal Templates described in the preceding item give page designers a great deal of flexibility, but you are still limited to the options there. HTML Templates allow you to use a third-party tool (like Dreamweaver) to create your OracleAS Portal pages. This method also gives you granular control of your OracleAS Portal page, down to the pixel. HTML Templates are also discussed in Chapter 4.

■ **Categories**  These allow you to organize your content for the benefit of your end users. Once a category is defined, a piece of content (an item) can be assigned to it. This way, end users can limit their search. An item can belong to only one category. The creation of a category is pretty straightforward and is not discussed in this book. For more on category definitions, see the sidebar “Defining Categories.”

■ **Navigation Pages**  These are just like subpages but have two distinct characteristics: 1) they are not searched for content when Oracle is told to index all content; 2) they have the Publish Page As Portlet box checked by default (meaning they show up in the Publish Portal Content category shown on Figure 3-13). As mentioned earlier, navigation pages are not really pages but rather elements designed to be placed on an existing page.

■ **Perspectives**  These are almost identical to categories except for the fact that an item can belong to multiple perspectives. As with categories, the creation of perspectives is trivial and not discussed in this book.

---

**Defining Categories**

While the creation of a category is simple, the definition of one is not. Many organizations struggle with this task when defining their portal. Great care must be taken so that categories and perspectives are created that are both functional and intuitive. Changing categories and perspectives after they have been defined and have had items associated with them is not an easy task. Likewise, adding a category or perspective and then updating all of the content in your portal can be arduous. Also, if categories and perspectives are not intuitive to your end users, their functionality will be severely crippled. It is imperative to do a thorough taxonomy definition before proceeding with the addition of content.
Styles, Attributes, Page Types, and Item Types

There are four other types of objects you can add to your page groups. Each of them provides the developer or page designer with additional ways of controlling how pages and the content on those pages are organized.

Styles

Many page designers confuse style and templates. A template can be used to define the structure of a page (how many regions, where they are placed, if a graphic is to be placed on the page, etc.), while a style is a set of attributes that can be applied to the elements on a page. The page elements are broken down into four basic categories:

- **Portlets** These affect how the portlets are displayed. Some of the portlet elements that can be modified include
  - **Portlet Header Color** The color of the horizontal bar on the top of the portlet
  - **Portlet Header Text** The color and font of the text that is in the portlet header
  - **Portlet Body Color** The background color of the portlet

- **Items** These affect how content is displayed. Some of the item elements that can be modified include
  - **Display Name** The font and color of the content name
  - **Group By Banner** The color, height, and alignment of text

- **Tabs** How active and inactive tabs are displayed

- **Common** How backgrounds and region banners are displayed

After a new style is defined (there is always a default style defined, named the `<Page Group Name>` style), it can be applied to a page by editing the page in Graphical mode, clicking the Style link on the top right of the page, and selecting the style to be used for the page. Page designers can also change the style for all pages in a page group by clicking the Properties link next to the Page Group heading in the top left of the page, selecting the Configure tab, selecting the Edit link in the Page Defaults section of the page, and changing the Default Style to the
newly created style. You can experiment by creating a new style and applying it to
the page without any fear of harming your page. You can always reassign your page
back to the original style without altering any of the regions, portlets, or content on
the page. The creation of a new style is as simple as clicking the Style link under the
Page Group, giving it a name, and setting the various properties for that style.

NOTE
If you create a style under a page group, that style
will be available only for pages in that page group. If
you want to create a style that can be shared among
page groups, return to the root of the page groups by
clicking the Page Groups link in the top left of the
screen. From there, you will see a link called Shared
Objects. When you select that link, you will see
links identical to those when you click one of the
Page Group links. Anything created here is available
to all page groups.

Attributes
Developers can add attributes to their pages to enhance the various attributes
already provided by OracleAS Portal. Predefined content attributes are associated
with item types and page types, and store information about an item or page, such
as the associated category, description, or perspectives. These attributes are
included in the Add and Edit screens, where users can provide information about
the item or page they are adding or editing. There is also another category of
attributes called display attributes. Display attributes 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.
Note that some content attributes, such as author and description, are also display
attributes. Any custom attributes created by the page group administrator are also
display attributes. Adding and using attributes is discussed in Chapter 5.

Page Types
OracleAS Portal comes with predefined page types. You can see a list of these by
going to the properties of a page group (by clicking the Properties link in the top left
of the graphical editing page of any page or by going to the Portal Navigator and
clicking the Properties link next to the Page Group in question), clicking the
Configure tab, and then clicking the Edit link in the Types And Classification section
of the page (Figure 3-16).
Here you see a list of the five predefined page types for OracleAS Portal:

- **Standard**  Creates a page that displays portlets and items. This is, by far, the most common page created in OracleAS Portal.

- **Mobile**  Creates a page that displays portlets and items. Mobile pages provide a way to design pages specifically for display in mobile devices, such as cell phones. Standard pages can also be displayed in mobile devices; however, the standard page design environment is more suited to creating pages for larger display devices, such as PC monitors.

- **URL**  Creates a page that displays the contents of a particular URL.
PL/SQL  Creates a page that displays the results of executing PL/SQL code.

JSP  Creates a page that displays the results of executing a JavaServer Page (JSP).

For most sites, these predefined page types are sufficient, but if you want to store more information than the default page types allow, you can create a custom page type. Custom page types are based on one of the existing base page types. The custom page type automatically inherits all of the base page type’s attributes. After you create a custom page type, you can edit it to add attributes that are specific to your requirements. It is important to note that you must create an attribute before you can add it to a custom page type. In addition to adding attributes to custom page types, you can also add calls to PL/SQL or HTTP procedures. You can pass attributes as parameters to the procedures if required, but you cannot add attributes or procedure calls to base page types.

Item Types
Just as it includes page types, OracleAS Portal comes with standard item types. For most sites, these item types are sufficient, but OracleAS Portal provides the mechanism for you to add your own item types if needed. Item types define the content of an item and the information that is stored about an item. Items in OracleAS Portal are based on item types. Items are one of the basic components of a portal page. The information stored about an item is determined by the attributes associated with the item type.

Base item types are the item types included with OracleAS Portal. There are two types of base item types. Content item types allow users to add content (for example, images, documents, or text) to a page. Navigation item types allow users to add navigational elements (for example, a login/logout link, a basic search box, or a list of objects) to a page. The base content item types are not actually available for users to add to pages. Instead, OracleAS Portal provides extended item types based on the base content items.

If the extended item types do not provide enough flexibility, you can further extend most of them to meet your requirements, provided you have the appropriate privileges. To extend these item types, you must have at least the page group privileges Manage Classifications and View on the Shared Objects page group. You can add different attributes to the item types to store exactly the information that you want. You can also add calls to PL/SQL and HTTP procedures and even pass attributes to the parameters of those procedures. You can only base custom item types on one of the base content item types. You cannot base custom item types on any of the base navigation item types.
Chapter 3: OracleAS Portal Page Design 143

Exercises

Most of the time, when OracleAS Portal Pages are created, page designers will want to create them from an existing template to ensure consistency. One of the really nice features of OracleAS Portal is the ability to convert a page into a template. The exercises in this chapter will focus on creating a page for our dashboard application. We will then convert that page into a template that can be used to create other pages.

Exercise 1: Edit the root page so that is has a basic look and feel for all of our subsequent pages. Take a good look at the page and see how it compares to other Web pages you have worked with, either internally in your company or on the Internet. You will spot a couple of obvious things to change:

- **The banner** The default banner has an image that says Oracle Application Server Portal. You’ll want to “brand” your site with your own logo. OracleAS Portal can accept images in the JPEG, GIF, or PNG format, and you can reference these images from either your local hard drive or a Web location. If you have an image you’d like to use already, modify the banner to use that. If you don’t have an image, you can download an image from the author’s site to your hard drive. The banner also has a title in the middle with the text “Dashboard_1 Banner.” Change that to something meaningful, or remove it altogether.

- **The style** The style of the page is rather plain. Change the style of the page group (or better yet, create a new style) and apply it to the page group. Make it so that it is pleasing to the eye yet captures the end user’s attention.

- **The regions** Most sites have a distinctive look and feel throughout all of their pages. As an example, go to Oracle’s technology Web site (http://www.oracle.com/technology/index.html). Almost all of the pages on this site have the same basic look and feel—a region on the left with headings like Products, Technologies, and Community; horizontal choices along the top like Downloads and Documentation; and the main part of the page that displays content (Figure 3-17). Not all of the pages have this layout, but by creating a template similar to this, new pages can be added quickly. Create regions in your template that use 25 percent of the screen on the left, 25 percent of the screen on the right, and 50 percent of the screen in the middle.

Exercise 2: Convert the page to a template by returning to the Portal Navigator and clicking the Convert Root Page to Template link. Save the template as Dashboard_Page_Template.
Exercise 3: Create a new page based on the Dashboard_Page_Template template.

Summary

The page design features of OracleAS Portal enable page designers to create a rich end-user experience quickly and easily. The template features (discussed more thoroughly in Chapter 4) take the hassle out of maintaining visual consistency throughout the site by giving page designers a simple way of generating and using templates. In the next chapter, advanced page design features are discussed, including using page parameters and events and—perhaps the most exciting new feature of OracleAS Portal 10.1.4—HTML Templates.