Oracle® Fusion Middleware

Planning a Portal with Oracle WebCenter Portal

11*g* Release 1 (11.1.1.9.0)

E35813-03

April 2015

Provides a starting point for portal development projects. Introduces portals and portal use cases at a high level, describes how to pick an appropriate development approach, and discusses the kinds of development tasks you'll need to plan for.

ORACLE®

Oracle Fusion Middleware Planning a Portal with Oracle WebCenter Portal, 11*g* Release 1 (11.1.1.9.0)

E35813-03

Contents

Preface

This guide provides a starting point for portal development projects. It introduces portals and portal use cases at a high level, describes how to pick the best development approach, and discusses the kinds of development tasks you'll need to plan for.

Audience

This document is intended for developers, application specialists, and advanced knowledge workers who plan and build portals. For a complete description of these roles, and other WebCenter Portal personas, refer to Section 1.4, "Building an Oracle WebCenter Portal Team: Who's Who?"

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle WebCenter Portal documentation set:

- *Using Oracle WebCenter Portal*
- *Building Portals with Oracle WebCenter Portal*
- *Administering Oracle WebCenter Portal*
- *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*
- *Installation Guide for Oracle WebCenter Portal*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started

This guide provides a starting point for portal development projects. It defines components and features, summarizes how to choose an appropriate portal development approach, and discusses the kinds of development tasks you'll need to plan for. Wherever possible, this guide links to background information and implementation details in the Oracle WebCenter Portal documentation set.

This chapter includes the following sections:

- Section 1.1, "What Is a Portal?"
- Section 1.2, "What Is Oracle WebCenter Portal?"
- Section 1.3, "Understanding Key Features in Oracle WebCenter Portal"
- Section 1.4, "Building an Oracle WebCenter Portal Team: Who's Who?"
- Section 1.5, "Navigating the Oracle WebCenter Portal Documentation Set"

1.1 What Is a Portal?

A *portal* offers secure online access to data, content, business applications, and services, presented in a web browser that is easy to navigate and use. Typically, portals present personalized information and content, and a unified view of applications and information aggregated from multiple back-end sources. Portals may also serve as a collaboration environment for users, whether they are employees, team members, customers, or business partners.

Businesses find many creative ways to use portals:

- Employees log into a company intranet portal to book travel, update payroll and tax forms, submit vacation requests, make hardware service requests, and perform many routine tasks.

- Human Resources managers add content to the portal for employee training sessions, important announcements, information targeted to new hires, insurance plan updates, and so on.

- Business partners and suppliers log into an extranet portal to assess inventory and manage orders.

Much of a portal's inherent value lies in the way it aggregates data, content, business applications, and services into a unified user experience, whether viewed in a browser, smart phone, or tablet. Figure 1–1 depicts several sources of information, exposed to a user on a single page in a portal.

Figure 1–1   Integrating Multiple Sources of Information
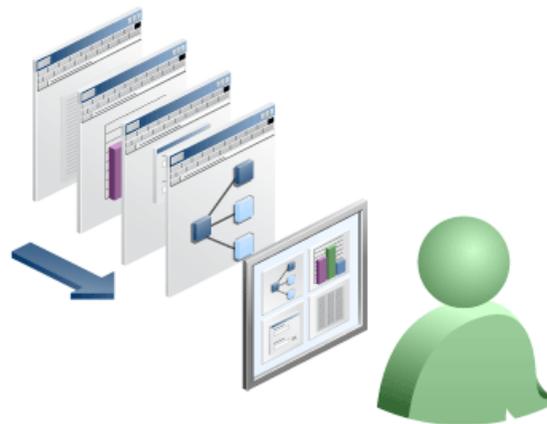
Portal users may be unaware that the data, content, and applications surfaced in a portal often come from entirely different back-end sources. Furthermore, through portal-driven events, these aggregated components can interact with each other within the browser. To the user, everything is presented in a seamless, integrated interface.

Portals falls into two basic categories:

- *Intranet portals* are used for collaboration, employee self-service, and company communication.

- *Extranet portals* are used by customers and partners for self-service and support.

What kind of portal do you want to build? Let's look at some of the basic requirements for these kinds of portals in more detail.

1.1.1 Internal Company Portals (Intranet)

Intranet portals allow users within an organization to collaborate, access critical information, and obtain self-service securely. There are many applications for an intranet portal, such as:

- A self-service portal that enables employees to update time cards, request vacation, complete performance appraisals, and view benefit information.

- Company-wide information or announcements that are made available to all users of the portal.

- Collaborative team workspaces, as described in Section 1.1.1.1, "Team Collaboration Portals."

1.1.1.1 Team Collaboration Portals

A team collaboration portal is typically used by a group within a company who wish to share and collaborate on documents and other information.

- Use a team portal for collaboration around a specific project or to present training content for all employees.

- You can make a team portal publicly available to users within the company or restricted to authorized users only, typically the members of the team or other invited individuals.

- Team portals may allow users to post, view, and edit documents.

- The collaboration portal might also include activity tracking, so users can see when a document was added or changed, among other activities.

Key features of team collaboration portals include:

- Team portals do not require any IT effort for creating, provisioning, and maintaining them. They are maintained primarily by knowledge workers and application specialists primarily. See Section 1.4, "Building an Oracle WebCenter Portal Team: Who's Who?" for information on these and other portal team roles.

- Portal users can share documents and use social components like activity streams, wikis, blogs, and so on. See Section 2.3, "Planning for Tools and Services."

- The system administrator delegates administration rights so that team leaders and other individuals can manage their own team portals, reducing the cost to the IT department and enabling them to serve large numbers of users with minimal support overhead. See Section 2.5, "Planning for Portal Security."

1.1.2 External Portals (Extranet)

Extranet portals extend outside the organization and provides services targeted to customers and vendors. Some extranet use cases include:

- Allowing authorized users to log in and access the portal's features. For example, vendors might access the site to learn about bidding opportunities or status. Customers can purchase products, view billing histories, or request support.

- Delivering custom application features to the portal as portlets, business mashups, or task flows.

- Presenting a consistent look and feel to users and emphasize corporate branding. Customer-specific branding may be required.

Typical requirements of intranet and extranet portals include:

- Well-defined methods for posting and updating content. See Section 2.2.1.3, "Integrating Content into a Portal with Portal Builder" and Section 2.2.2.3, "Integrating Content into a Portal with Portal Framework."

- Access to applications from multiple sources that may be provided through out-of-the-box or custom components like portlets, task flows, and mashups; collaboration and workflow features. See Section 1.3.2, "Integration with Other Oracle Products and Applications" and Section 2.3, "Planning for Tools and Services."

- Consistent branding and look and feel. See Section 2.4, "Planning for a Consistent Look and Feel."

- Resources are secured and only authorized users can access the resources. See Section 2.5, "Planning for Portal Security."

- Secure access to applications from multiple sources (single sign-on). See Section 2.5, "Planning for Portal Security."

1.2 What Is Oracle WebCenter Portal?

Companies use *Oracle WebCenter Portal* to build enterprise-scale intranet and extranet portals. Portals built with Oracle WebCenter Portal commonly support thousands of users who create, update, and access content and data from multiple back-end sources. From the user's perspective, the integration is seamless.

Figure 1–2 shows the basic structure of a portal in Oracle WebCenter Portal, showing the initial landing page, named Personal. It includes branding in the header region with a company logo and a **Sign In** link, navigation areas, content areas, and so on.

Figure 1–2   Sample Portal Built with Oracle WebCenter Portal
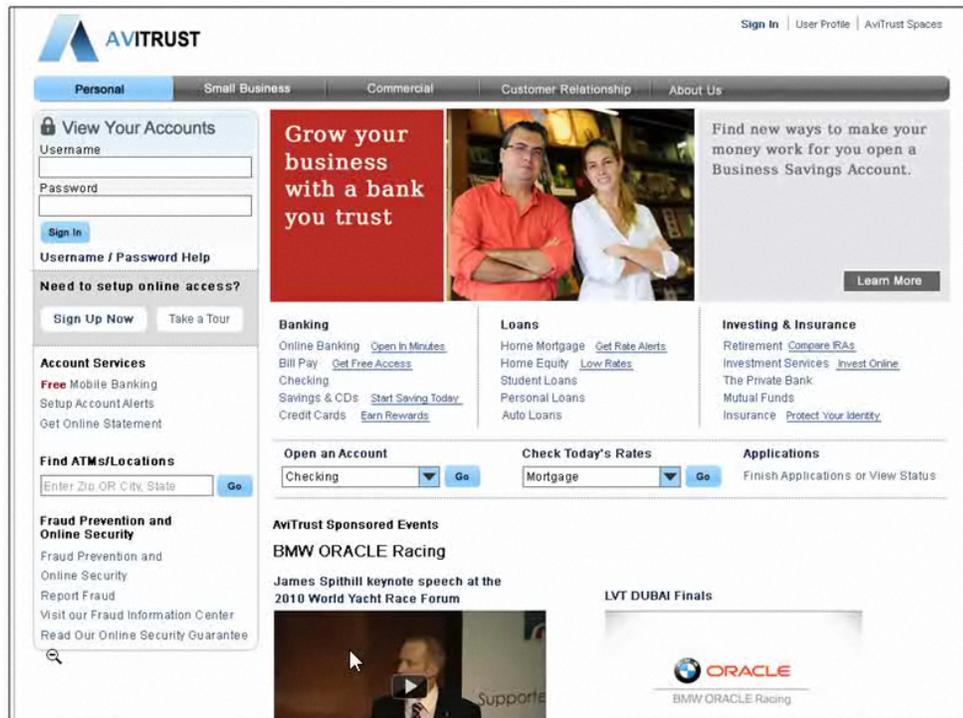
1.3 Understanding Key Features in Oracle WebCenter Portal

Oracle WebCenter Portal provides an open, secure, standards-based, and extensible framework that allows you to rapidly create and deploy portals. This section briefly describes some of Oracle WebCenter Portal's key features:

- Section 1.3.1, "Portal Security"

- Section 1.3.2, "Integration with Other Oracle Products and Applications"

- Section 1.3.3, "Integration with Oracle WebCenter Content"

- Section 1.3.6, "Search"

- Section 1.3.4, "Tools and Services"

- Section 1.3.7, "Device Support"

- Section 1.3.8, "Portlets"

- Section 1.3.9, "Pagelets"

- Section 1.3.10, "Localized Content and Language Support"

- Section 1.3.11, "Portal Life Cycle"

- Section 1.3.5, "Assets and Round-Trip Asset Development"

1.3.1 Portal Security

When you create a new portal, you can set certain security policies that define who can access it. Typically, portals require users to authenticate themselves by logging in. Access to portal pages and components can be specified through delegated administration. Delegated administration refers to assigning management rights to portions of a portal, such as specific page hierarchies, to other individuals so that administration tasks can be distributed to appropriate managers.

For more information, see:

- Section 2.5, "Planning for Portal Security"
- "Understanding Security" in *Administering Oracle WebCenter Portal*.
- "Administering Security in a Portal" in *Building Portals with Oracle WebCenter Portal*
- "Granting Users Access to a Portal" in *Building Portals with Oracle WebCenter Portal*.

1.3.2 Integration with Other Oracle Products and Applications

To provide a streamlined user experience, Oracle WebCenter Portal is integrated and certified with many Oracle technologies and applications.

Other technologies that can be used with or integrated into a portal are Oracle Fusion Applications, Oracle Identity Management Suite, Oracle SOA Suite, Oracle Enterprise Manager, and Oracle eBusiness Suite, as well as PeopleSoft, Siebel, and JD Edwards enterprise applications.

For more information, see "Integrating Other Oracle Applications" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.

1.3.3 Integration with Oracle WebCenter Content

Oracle WebCenter Portal integrates with Oracle WebCenter Content, which provides a flexible, secure, centralized, web-based repository for managing all phases of the content life cycle: from creation and approval to publishing, searching, expiration, and archival or removal.

When authorized users upload content to a portal, for example, the content can be stored and managed in Oracle WebCenter Content, which can store multiple versions, be used for archiving, manage approval processes and workflows, and create different renderings.

Oracle WebCenter Portal's Content Presenter feature enables you to precisely customize the selection and presentation of content in a portal. Content Presenter connects to and is closely integrated with Oracle WebCenter Content.

For more information, see:

- Section 2.2.1.3, "Integrating Content into a Portal with Portal Builder"
- Section 2.2.2.3, "Integrating Content into a Portal with Portal Framework"
- "Managing Content Repositories" in *Administering Oracle WebCenter Portal*
- "Working with Documents" in *Using Oracle WebCenter Portal*.
- "Introduction to Working with Content" in *Building Portals with Oracle WebCenter Portal*
- "Publishing Content Using Content Presenter" in *Building Portals with Oracle WebCenter Portal*

- "Working with Document Task Flows and Document Components" in *Building Portals with Oracle WebCenter Portal*

1.3.4 Tools and Services

Oracle WebCenter Portal includes a wide range of tools and services that you can leverage in a portal. These tools and services include mail, document and content management, discussion forums, calendars, wikis, blogs, tagging, activity tracking, personalization, polls, instant messaging, and more.

In most cases, some back-end configuration is required. For example, to use discussions in a portal, the discussions server must be running and a back-end connection to the server configured. These implementation tasks are described in detail in the Oracle WebCenter Portal documentation.

For more information, see:

- Section 2.3, "Planning for Tools and Services"
- "Introduction to Portal Tools and Services" in *Building Portals with Oracle WebCenter Portal*.

1.3.5 Assets and Round-Trip Asset Development

Assets are the objects that define the structure, look and feel, and the content of portals. WebCenter Portal provides the following assets, which can be used out-of-the-box or customized: page templates, resource catalogs, skins, page styles, Content Presenter display templates, task flow styles, pagelets, task flows, and data controls.

Round-trip asset development refers to features and techniques that allow you to retrieve portal assets from a previously deployed portal and pull them back into JDeveloper for maintenance or enhancement. After modifying an asset in JDeveloper, you can upload the asset back to the deployed portal. Round-trip development features provide a simple, convenient way to modify portal assets without redeploying the entire application.

For more information about how WebCenter Portal supports round trip development, see:

- Section 2.2.1.2, "Managing Portal Assets with Portal Builder"
- Section 2.2.2.2, "Managing Portal Assets with Portal Framework"

1.3.6 Search

Oracle Secure Enterprise Search (Oracle SES) is the default search platform for Oracle WebCenter Portal. Oracle SES provides excellent scalability and performance. In addition, it provides unified ranking results and search of external repositories.

For more information, see:

- "Adding Search to a Portal" in *Building Portals with Oracle WebCenter Portal*
- "Integrating Search" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*
- "Managing Oracle Secure Enterprise Search in WebCenter Portal" in *Administering Oracle WebCenter Portal*

1.3.7 Device Support

Out-of-the-box, Oracle WebCenter Portal is supported on a number of popular browsers, smart phones, and tablets. In addition, Oracle WebCenter Portal allows you to add support for these devices through the use of mobile device templates.

For more information, see "Administering Device Settings" in *Administering Oracle WebCenter Portal*.

1.3.8 Portlets

Standards-based portlets provide a means of presenting data from multiple sources in a meaningful and related way. Portlets can display excerpts of other web sites, generate summaries of key information, perform searches, and access assembled collections of information from a variety of data sources. Because several different portlets can be placed on a single page, users benefit from a single-source experience even though, in reality, the content may be derived from multiple sources.

For more information, see:

- "Introduction to Portlets" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*

- "Managing Portlet Producers" in *Administering Oracle WebCenter Portal*

1.3.9 Pagelets

A pagelet is a reusable user interface component. Any HTML fragment can be a pagelet, but pagelet developers can also write pagelets that are parameterized and configurable, to dynamically interact with other pagelets, and respond to user input. Pagelets can be run on any web page, including within a portal or other web application. Pagelets can be used to expose platform-specific portlets in other web environments. Oracle WebCenter Portal's Pagelet Producer provides a collection of useful tools and features that facilitate dynamic pagelet development.

For more information, see:

- "Creating Pagelets with Pagelet Producer" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*

- "Managing the Pagelet Producer" in *Administering Oracle WebCenter Portal*

1.3.10 Localized Content and Language Support

If your portal will be viewed by users in more than one country, you can configure it to display localized content based on the user's selected language and locale.

Additionally, locale selection applies special formatting considerations applicable to the selected locale. For example, those considerations may include whether information is typically viewed from left to right or right to left, how numbers are depicted (such as monetary information), and so on.

For more information, see "Choosing a Default Display Language" in *Administering Oracle WebCenter Portal*.

1.3.11 Portal Life Cycle

The portal life cycle refers to the path a portal takes from development through production. The phases of the life cycle typically include development, testing, staging, and production. Each phase requires certain tasks to be performed. Some

tasks are performed only once in a given environment, like setting up a content repository. Others are performed more frequently, like updating content.

Usually, you will install and configure back-end services required by your portal and applications used by the portal in each environment. For example, the staging environment might include a dedicated content server, such as Oracle WebCenter Content as well as a dedicated portlet producer server (`WC_Portlet`), a utilities server for analytics, activity graph, data integration (`WC_Utilities`), and a collaboration server for discussions and announcements (`WC_Collaboration`). The staging server is often maintained as a mirror of the production site. For more information and detailed diagrams, see, the "Oracle WebCenter Portal Topology" section in *Administering Oracle WebCenter Portal*.

Part of your portal planning process is to understand how to move portals, as well as their assets, data, and content from one environment to another. For more information, see:

- Section 2.2.1.1, "Understanding the Portal Life Cycle with Portal Builder"
- Section 2.2.2.1, "Understanding the Portal Life Cycle with Portal Framework"

1.4 Building an Oracle WebCenter Portal Team: Who's Who?

The Oracle WebCenter Portal product and documentation have been carefully designed around *user personas*. The idea is that certain users perform specific kinds of tasks, generally, depending on their role. An understanding of these personas, and the kinds of tasks they typically perform, may help guide you in building a team that will create and manage portals across your enterprise.

Here's a quick summary of the personas, the tasks they generally perform, and the tools they use:

- **Knowledge Worker**. Knowledge workers focus on viewing and possibly adding content to a portal, and reviewing the content of others. Occasionally, knowledge workers may create a new portal based on existing templates for team collaboration or other in-house use.

- **Application Specialist**. Application specialists create and administer portals, build portal templates and page templates, and add content, services, and components to them. For example, an application specialist might create a team collaboration portal template that includes user profiles, a wiki, a documents section, and other useful features. That template could then be used by knowledge workers on individual teams who create and manage their own team portals.

  Application specialists are typical portal creators, and thus by default are the administrator of the portals they create. In this role (known as the portal moderator), they can establish portal security, such as delegate portal creation and management rights to other application specialists. These users, in turn, can delegate certain rights to other users, such as knowledge workers responsible for updating content on a daily basis. Furthermore, collaborative teams can be formed that grant rights to team members who contribute and collaborate on content using their team portal. The ability to delegate portal administration rights across a spectrum of possible users greatly decreases the overall cost to the IT department, enabling thousands of users to be served with minimal support overhead.

- **Web Developer**. Web developers deliver a consistent, branded look and feel to all portals. A web developers works with graphic design tools to define proposals, then either passes approved designs to the application specialist to implement as

templates, or edits page templates directly. Advanced web developers use technologies like JavaScript, HTML, Ajax, JSP, and so on to create rich, interactive user interfaces.

- **Developer**. Portal developers work primarily in JDeveloper to create custom portal components, like portlets, task flows, and managed beans. Developers create components that require Java, SQL, Expression Language, Oracle Application Development Framework (ADF), and other programming skills.

- **System Administrator**. System administrators manage and set global options for all portals, rather than administering individual portals. As needed, a system administrator configures back-end systems such as content repositories, databases, services, and applications. Many portal components rely on these connections, including document components, components like discussions and activity tracking, and others. They also manage portal backups and recovery. A system administrator typically works with portal administration tools, Oracle Enterprise Manager, and command line tools like WebLogic Scripting Tool (WLST). For a good introduction to system administration, see "Introduction to Administering Oracle WebCenter Portal" in *Administering Oracle WebCenter Portal*.

For more detailed information, including which guides are most suitable for each persona, see the "Who's Who" chapter, which appears at the beginning of each of the Oracle WebCenter Portal guides.

1.5 Navigating the Oracle WebCenter Portal Documentation Set

This section lists the primary guides for Oracle WebCenter Portal and their intended audiences, as outlined in Section 1.4, "Building an Oracle WebCenter Portal Team: Who's Who?"

- *Administering Oracle WebCenter Portal*

  This guide is aimed at system administrators. It explains how to manage global aspects of Oracle WebCenter Portal. System administrators learn how to set up, run, and administer Oracle WebCenter Portal's browser-based portal development environment, as well as any portal applications built using JDeveloper. This guide also covers how to deploy portals, configure back-end services, monitor performance, implement portal backup and recovery, and more.

- *Building Portals with Oracle WebCenter Portal*

  This guide is aimed at the application specialist and advanced knowledge worker who creates, edits, and administers portals. It provides instructions for tasks such as creating, customizing, and administering portals and subportals; managing and developing portal templates; creating and editing portal pages and the content they expose; defining the assets and tools available to portals; setting portal and page security through membership, roles, and permissions; and more.

- *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*

  This guide is aimed at the developer who creates Portal Framework applications, portlets, and custom portal components using Oracle JDeveloper. Many of the tasks described in this guide involve activities that require Java, CSS, Application Development Framework (ADF), Expression Language (EL), and related experience.

- *Using Oracle WebCenter Portal*

  This guide is aimed at the knowledge worker and other end users, who interact with information and users through WebCenter Portal. It provides instructions for

tasks such as working in a personal environment in the Home portal; working in a portal as a portal member; finding and updating content; creating new content; collaborating with other business users to create, edit, and share documents; and sending and receiving instant messages with peers.

Other documents cover installing and configuring Oracle WebCenter Portal. Typically, these activities are handled by a system administrator. For detailed information, see:

- *Portal Installation and Configuration Roadmap*

- *Quick Installation Guide for Oracle WebCenter Portal*

- *Installation Guide for Oracle WebCenter Portal*

- *Patching Guide*

2

Planning and Building a Portal

As a portal architect or designer, your first challenge is to choose the best fundamental development approach. As you will learn in this chapter, this choice will guide the overall direction and implementation of your portal project.

This chapter includes the following sections:

- Section 2.1, "Portal Planning Overview"
- Section 2.2, "Planning for Building a Portal"
- Section 2.3, "Planning for Tools and Services"
- Section 2.4, "Planning for a Consistent Look and Feel"
- Section 2.5, "Planning for Portal Security"
- Section 2.6, "Post-Planning Tasks Summary"

2.1 Portal Planning Overview

If you are building an intranet portal to support multiple lines of business, an extranet portal to support customers and business partners, a team collaboration portal, or if you have other portal use cases, you will need to address these basic planning tasks:

- **Assemble your portal team.**

  Remember that Portal Builder is primarily a browser-based approach that is oriented toward knowledge workers and application specialists. Portal Framework is primarily a developer-centric approach that requires JDeveloper skills to create and evolve the portal.

  See Section 1.4, "Building an Oracle WebCenter Portal Team: Who's Who?"

- **Decide on your development approach.**

  This decision guides most aspects of your portal implementation, the composition of your portal development team, and certain features that are available to you.

  See Section 2.2, "Planning for Building a Portal."

- **Plan for portal content integration.**

  No matter what kind of portal you create, you will need a content integration and management strategy. Oracle WebCenter Portal provides several ways to integrate content, including the document task flows (such as the Document Explorer) and Content Presenter. Document-related features require a connection to a content repository. Typically, content connections are configured by a system administrator.

See Section 2.2.1.3, "Integrating Content into a Portal with Portal Builder" and Section 2.2.1.3, "Integrating Content into a Portal with Portal Builder."

- **Identify the tools and services your portal requires.**

  WebCenter Portal offers many tools and services that allow portal members to collaborate and communicate. For example, for a company intranet portal, announcements are useful for spreading information of interest to a like-minded group, immediately or on a selected date and time. Activity graphs are useful for providing recommendations for pages and documents of interest based on an analysis of user activity. The instant messaging and presence tool keeps department members informed about each other's online status and provides options for contacting another user on the spot. Worklists are useful for keeping track of workflow notifications and messages channeled through the User Messaging Service (UMS). See Section 2.3, "Planning for Tools and Services."

- **Define the basic organizational scheme for your portal.**

  There are many ways to organize a portal. For example, your portal might reflect your company's departmental organization, lines of business, or business processes.

  All users could have access to the central landing page, which might be a Home page, the Home portal, or the parent portal in a set of hierarchical portals. You can also use role-based navigational elements and delegated role-based access through the page hierarchy.

  See Section 2.4, "Planning for a Consistent Look and Feel."

- **Design a consistent layout and look for your portal pages.**

  Portal layout is controlled primarily with page templates and page styles. Page templates determine the overall organization of pages within the portal. Templates typically include branding elements, like the company logo, that are intended to appear on all pages. Page styles allow you to standardize how content is organized within pages. Skins define colors, fonts, and other "look and feel" aspects of various components used on the pages of a portal.

  Developers can create custom templates and skins in JDeveloper that can uploaded to a deployed portal, or knowledge workers can create templates by copying and modifying existing templates.

  See Section 2.4, "Planning for a Consistent Look and Feel" and Section 1.3.5, "Assets and Round-Trip Asset Development."

- **Define security needs your portal, its pages, and resources.**

  See Section 2.5, "Planning for Portal Security" for an overview of portal security and for links to Oracle WebCenter Portal documentation for implementation information.

- **Decide which applications may need to be integrated into the portal.**

  Other technologies that can be used with or integrated into a portal are Oracle Fusion Applications, Oracle Identity Management Suite, Oracle SOA Suite, Oracle Enterprise Manager, and Oracle eBusiness Suite, as well as PeopleSoft, Siebel, and JD Edwards enterprise applications, and others. See Section 1.3.2, "Integration with Other Oracle Products and Applications."

  Portlets and pagelets provide another way to bring remote applications and content into a portal. See Section 1.3.8, "Portlets" and Section 1.3.9, "Pagelets."

The main Oracle WebCenter Portal documentation set provides implementation details on all of the features described in this guide, and more. See Section 1.5, "Navigating the Oracle WebCenter Portal Documentation Set."

2.2 Planning for Building a Portal

The first important design decision to make is to choose a fundamental development approach for your portal. Oracle WebCenter Portal 11*g* Release 1 allows two distinctly different approaches: *Portal Builder* and *Portal Framework application* (see Figure 2–1). Most implementation, administration, and life cycle tasks for a portal project depend on which approach you choose.

> **Note:** Oracle recommends creating new portals using Portal Builder, rather than Portal Framework. While custom Portal Framework applications may provide a greater degree of flexibility when developing your portal solution, such flexibility typically forfeits product supportability and the ability to upgrade to new WebCenter Portal releases. Using Portal Builder avoids these constraints without sacrificing complexity or scalability.

Figure 2–1   Planning for Building a Portal
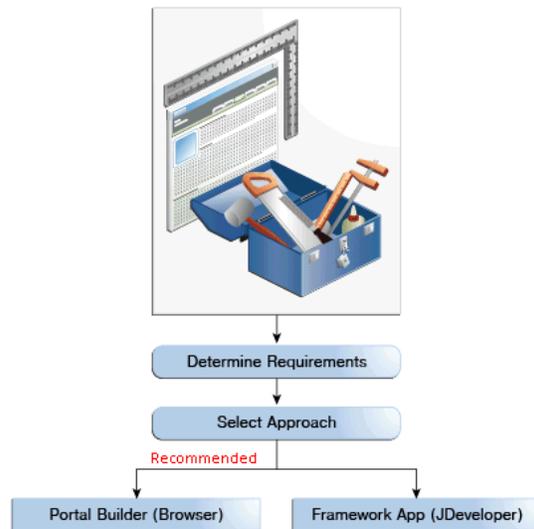
This section discusses both Portal Builder and Portal Framework approaches in more detail:

- Section 2.2.1, "Considering the Portal Builder Approach"
- Section 2.2.2, "Considering the Portal Framework Approach"
- Section 2.2.3, "Comparing Portal Builder and Portal Framework Approaches"

2.2.1 Considering the Portal Builder Approach

Portal Builder is a browser-based, packaged environment for building portals in WebCenter Portal. Oracle recommends Portal Builder as a long-term solution to your portal needs.

In Portal Builder, you can create a portal, add and edit the pages of a portal in the page editor (*Composer*), and administer a single portal as the portal creator. The system administrator has access to the Portal Builder administration area that allows for administering all portals. Using Portal Builder, portal creators select from out-of-the-box or custom portal and page templates, simplifying and standardizing the portal creation process. A drag-and-drop paradigm makes it easy to add content, layout, and feature components to the portal.

Portal Builder is oriented toward *application specialists* and *knowledge workers* who create, build out, customize, and manage individual portals, and *system administrators* who manage global portal settings. Figure 2–2 shows the Portals page in Portal Builder in a browser.

Figure 2–2   Creating a New Portal in Portal Builder
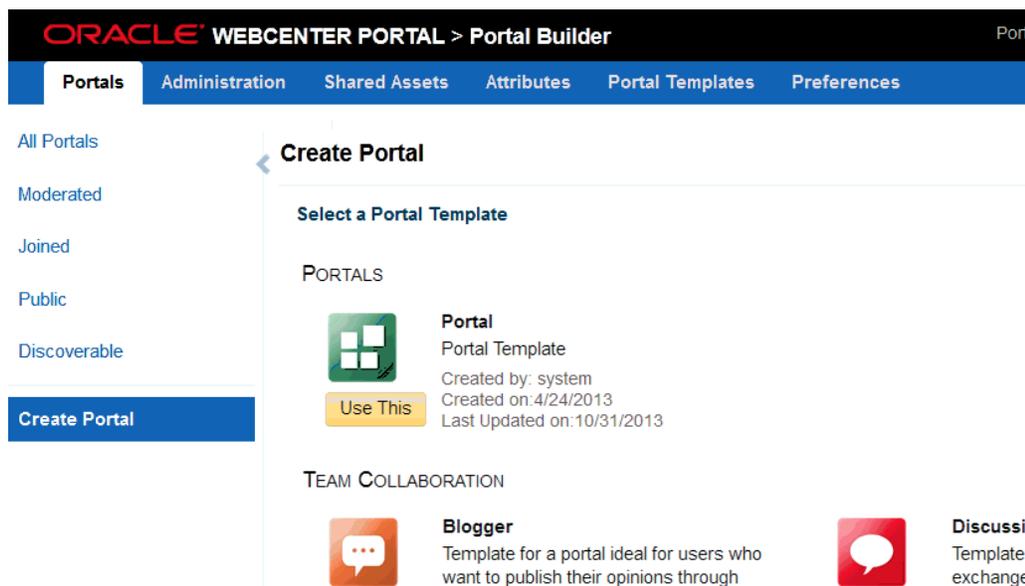

When you choose the Portal Builder path, developer skills, such as ADF and Java, are only required if you want to develop custom components that are not provided by Oracle WebCenter Portal out-of-the-box. Developers can use JDeveloper to build and deploy any specialized custom skins, page styles, page templates, content display templates, navigations, resource catalogs, data controls, task flows, portlets, and managed beans that you may want to offer in Portal Builder.

It is important to note that Portal Builder is a *multi-portal* solution. You can install a single instance of Oracle WebCenter Portal and host multiple portals within it. You can use out-of-the-box templates to get started and subsequently customize them. Typically, system administrators delegate administration rights to designated individuals, so they can manage the portal themselves, reducing the cost to the IT department and enabling thousands of users to be served with minimal support overhead. Portal Builder is the only approach that provides built-in mobile device support and capabilities.

Everything you need to know about building and managing portals with Portal Builder is documented in *Building Portals with Oracle WebCenter Portal*. Tasks for administering the Portal Builder environment are covered in "Managing Portals in Portal Builder Administration" in *Administering Oracle WebCenter Portal*. The following sections provide a summary overview:

- Section 2.2.1.1, "Understanding the Portal Life Cycle with Portal Builder"

- Section 2.2.1.2, "Managing Portal Assets with Portal Builder"

- Section 2.2.1.3, "Integrating Content into a Portal with Portal Builder"

- Section 2.2.1.4, "Administering Oracle WebCenter Portal with Portal Builder"

2.2.1.1 Understanding the Portal Life Cycle with Portal Builder

The portal life cycle describes the process of creating a portal using Portal Builder through deployment to a production instance. The phases of the life cycle typically include development, testing, staging, and production. Each phase requires certain tasks to be performed. Some tasks are performed only once, like setting up a content repository. Others are performed more frequently, like creating backups, and performing nightly builds. With Portal Builder, it is easy to export an entire portal or individual assets and import them into a different environment.

Figure 2–3 illustrates the general flow from staging to production environments.

Figure 2–3   Flow from WebCenter Portal Staging to Production Environments
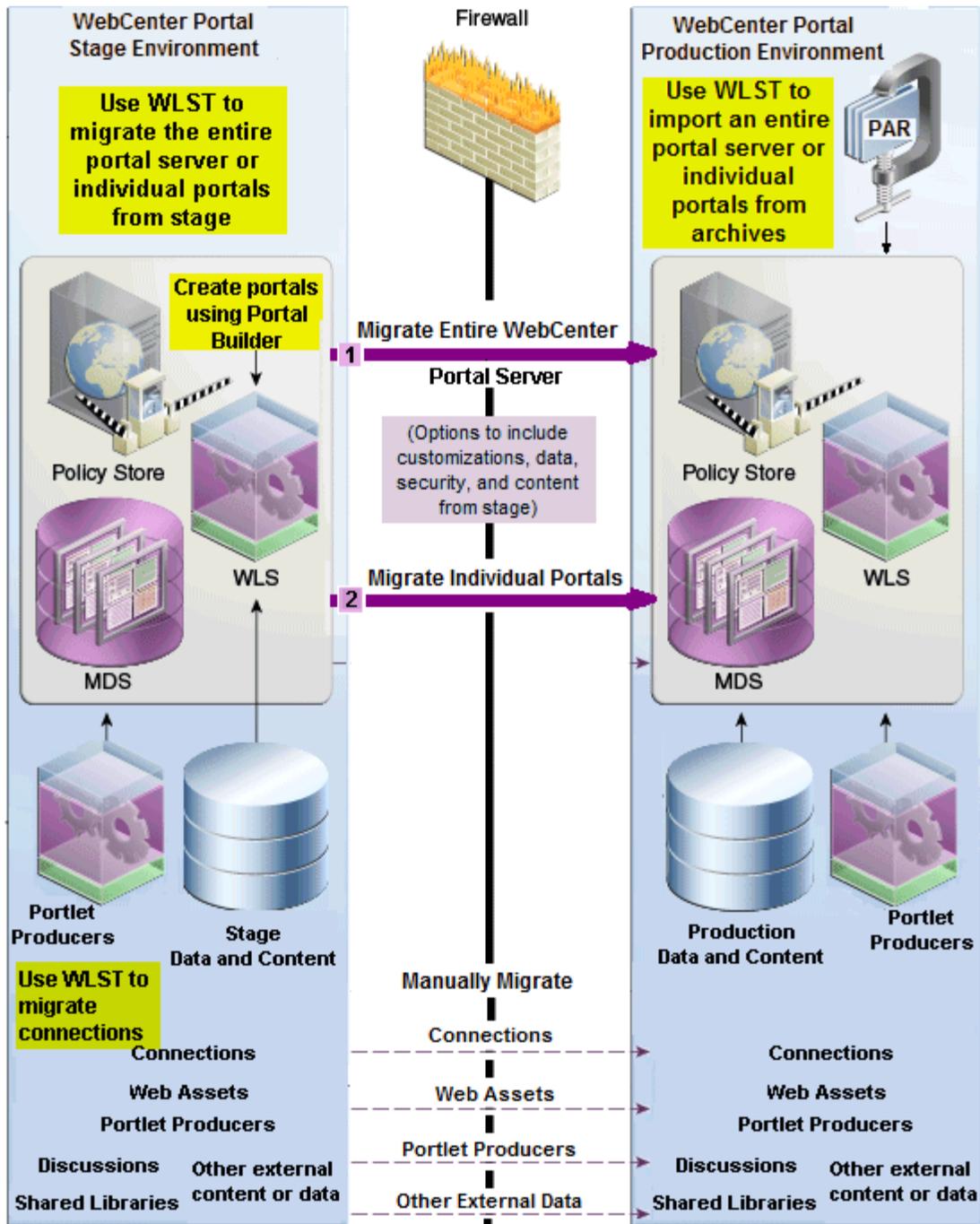
For more information, see:

- "Understanding WebCenter Portal Life Cycle" in *Administering Oracle WebCenter Portal*.

- "Deploying Portals" in *Administering Oracle WebCenter Portal*.

2.2.1.2 Managing Portal Assets with Portal Builder

Assets are the objects that define the structure, look and feel, and the content of portals. WebCenter Portal provides the following assets, which can be used

out-of-the-box or customized: page templates, resource catalogs, skins, page styles, Content Presenter display templates, task flow styles, pagelets, task flows, and data controls.

If your portal asset requirements extend beyond the browser-based editing capabilities of Portal Builder, developers can further develop assets in JDeveloper from within a special project called `WebCenterSpacesResources`. This project is part of a JDeveloper workspace that you can download from Oracle Technology Network. The JDeveloper project provides everything developers need to create, modify, and upload assets like page templates, skins, page styles, resource catalogs, and others.

JDeveloper supports the direct uploading and downloading of assets to and from the `WebCenterSpacesResources` project and Portal Builder. You can also export an asset to a file and then import it into Portal Builder or JDeveloper.

For more information, see:

- "Developing Assets for WebCenter Portal" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*

- "Deploying Assets" in *Administering Oracle WebCenter Portal*

2.2.1.3 Integrating Content into a Portal with Portal Builder

One of the most important portal requirements is content integration. Portal creators and managers need to add and update content, manage content creation, workflows, archiving, and so on.

A *content repository* stores content such as documents and images. A connection to a back-end content repository (Oracle WebCenter Content) is created for Portal Builder out-of-the-box. Once this connection is available, portal moderators can enable the documents tool in their portals from the Tools and Services page of Portal Builder. They can then add document components to portal pages, and end users can work with the documents in the portal.

For more information, see:

- "Managing Content Repositories" in *Administering Oracle WebCenter Portal*.

- "Working with Documents" in *Using Oracle WebCenter Portal*.

- "Enabling and Disabling Tools and Services Available to a Portal" section in *Building Portals with Oracle WebCenter Portal*

- "Publishing Content Using Content Presenter" in *Building Portals with Oracle WebCenter Portal*.

- "Working with Document Task Flows and Document Components" in *Building Portals with Oracle WebCenter Portal*.

2.2.1.4 Administering Oracle WebCenter Portal with Portal Builder

The system administrator is the only user assigned to the WebCenter Portal `Administrator` role. WebCenter Portal administrators have the highest application privileges within the WebCenter Portal application (which includes Portal Builder). The system administrator can view and customize every aspect of WebCenter Portal, manage users and roles, and delegate responsibilities to others.

An individual portal is managed by the user who creates the portal (assigned the portal `Moderator` role) or a portal member with `Manage All` or `Manage Configuration` permission in the portal. Users are assigned portal permissions and roles through the portal administration security settings.

A user who accesses administration settings for a portal and does not have `Manage` permission (for example, a user with `Participant` role, granted permission to edit a page in the portal) will see only the settings available to their role and permissions.

Figure 2–4 shows the Portals page in Portal Builder. This page displays a table showing all the available portals. If you have one of the `Manage` permissions for a portal, an **Administer** link shows next to that portal, which allows you to access the administration settings for the portal.

Figure 2–4   Portal Builder Portals Page
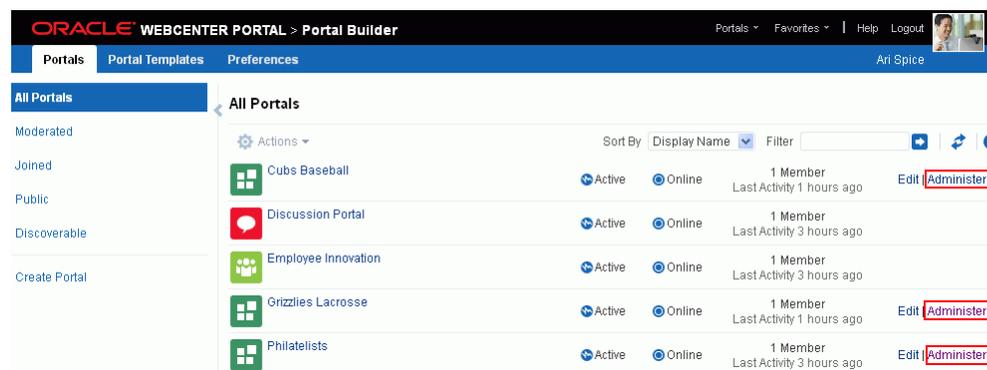
For more information, see:

- "Administering a Portal" in *Building Portals with Oracle WebCenter Portal*.

- "System Administration for WebCenter Portal" and "System Administration for Portal Builder" in *Administering Oracle WebCenter Portal*.

2.2.2 Considering the Portal Framework Approach

> **Note:**   Oracle recommends creating new portals using Portal Builder, rather than Portal Framework. While custom Portal Framework applications may provide a greater degree of flexibility when developing your portal solution, such flexibility typically forfeits product supportability and the ability to upgrade to new WebCenter Portal releases. Using Portal Builder avoids these constraints without sacrificing complexity or scalability.

A *Portal Framework application* is a highly customizable approach to building a portal. As the more complex approach, portals built using Portal Framework require considerable Oracle ADF and Java development expertise and require using Oracle JDeveloper, Oracle's Java development environment. Because Portal Framework applications are pure ADF applications, they integrate well with other ADF applications, which could be important if you have such applications. Figure 2–5 shows a portal project open in JDeveloper.

Figure 2–5 A Portal Project in JDeveloper
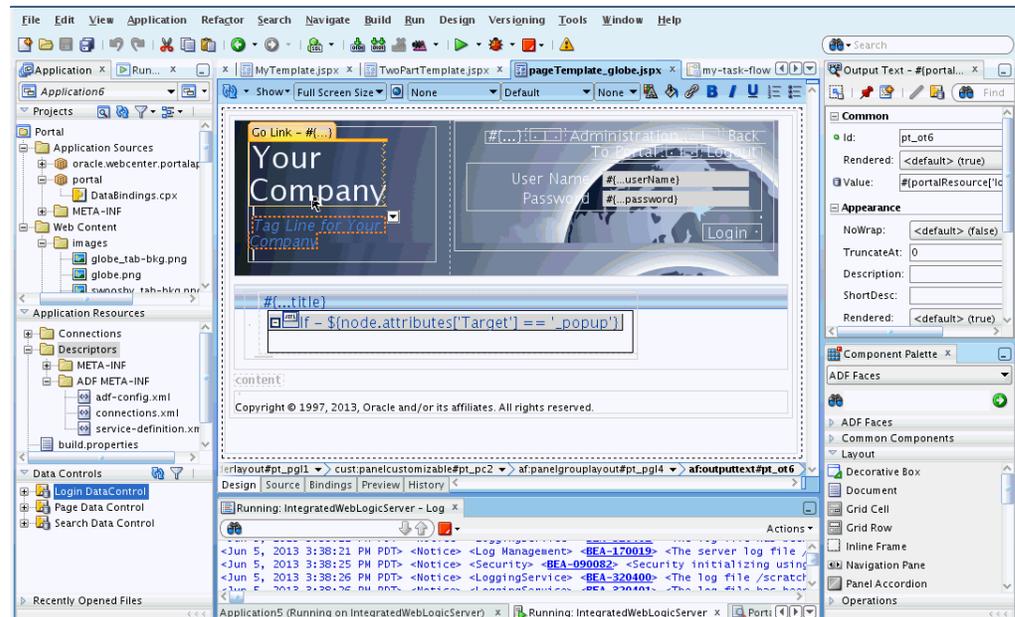
The Portal Framework approach typically requires a dedicated team of *developers* who create both the portal and any custom components that may be needed. A browser-based portal administration environment is provided that allows administrators and others to manage and customize a portal once it is deployed to a server. However, it is important to understand that a Portal Framework application is primarily a developer-focused approach.

Unlike Portal Builder, Portal Framework is a *single-portal approach*. It is not possible to create multiple portals or a portal hierarchy (subportals) within a Portal Framework application.

Everything you need to know about building Portal Framework applications is covered in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*. The following sections provide a summary overview:

- Section 2.2.2.1, "Understanding the Portal Life Cycle with Portal Framework"

- Section 2.2.2.2, "Managing Portal Assets with Portal Framework"

- Section 2.2.2.3, "Integrating Content into a Portal with Portal Framework"

- Section 2.2.2.4, "Administering Portal Framework Applications"

2.2.2.1 Understanding the Portal Life Cycle with Portal Framework

The WebCenter Portal Framework application life cycle refers to the path a Portal Framework application takes from development through production. The phases of the life cycle typically include development, testing, staging, and production. Each phase requires certain tasks to be performed. Some tasks are performed only once, like setting up a content repository. Others are performed more frequently, like nightly builds.

If you build a Portal Framework application, you need to consider setting up a team development environment, a build environment, and environments for testing, staging, and finally, production.

Developers and administrators typically use standard web application deployment methods (EAR deployment), WLST (WebLogic Scripting Tool), Enterprise Manager,

and an out-of-the-box propagation tool to move portals, portal assets, and portal customizations from one environment to another. JDeveloper allows developers to deploy portals quickly and iteratively to an internal server for testing.

Figure 2–6 illustrates the general flow from development to build to test environments.

Figure 2–6   Flow from Development to Build to Test Environments
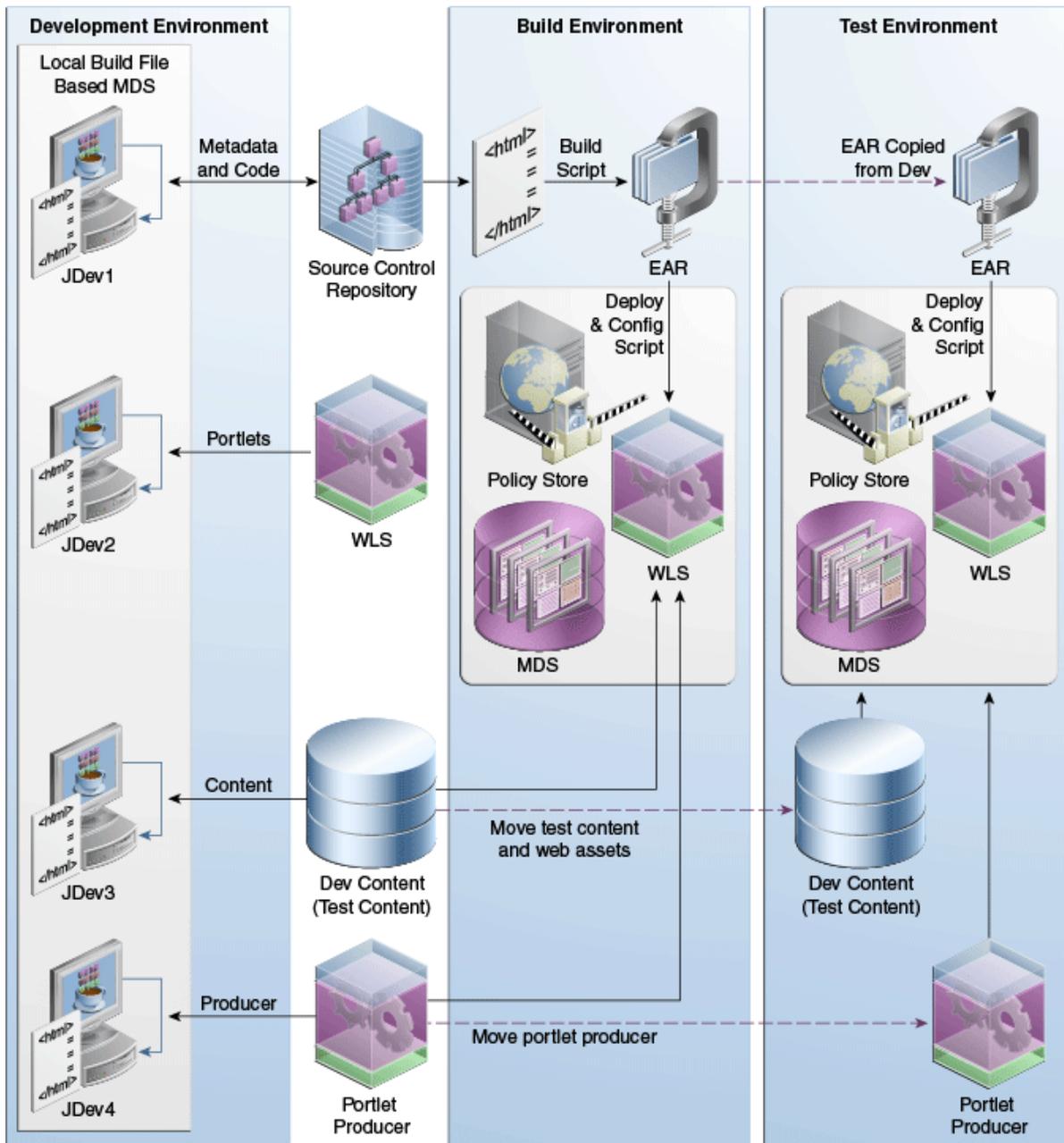
For more information, see the following chapters in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*:

- "Working Productively in Teams"

- "Understanding the WebCenter Portal Framework Application Life Cycle"

- "Deploying and Testing Your Portal Framework Application"

2.2.2.2 Managing Portal Assets with Portal Framework

In a Portal Framework application, round-trip development refers to features and techniques that allow you to retrieve assets (such as page templates or skins) from a previously deployed portal and pull them back into JDeveloper for maintenance or enhancement. After modifying an asset in JDeveloper, you can use a feature called the Resource Manager to upload the resource back to the deployed portal.

The typical pattern used for round-trip development of assets in a Portal Framework application is to download the asset from the deployed portal to a file (using the Assets page of the Administration Console), then import the downloaded file into JDeveloper and work on it. Then, you export it from JDeveloper and upload the exported asset back to the deployed portal using the Assets page of the Administration Console.

For more information, see:

- "Working with Round-Trip Development" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.

- "Managing Assets for a Portal Framework Application" in *Administering Oracle WebCenter Portal*.

2.2.2.3 Integrating Content into a Portal with Portal Framework

A *content repository* stores content such as documents and images. A developer can create content repository connections directly in JDeveloper. These connections allow the Portal Framework application's content components to access and display content while the developer is actively working on the portal in JDeveloper. The development team may install and maintain its own content repository instance for their exclusive use.

For portals in a staging or production managed server environment, content repository connections are typically configured and maintained by a system administrator.

After content repository connections are made, you can integrate content into a Portal Framework application in a variety of ways. For example, you can use the Document Explorer (shown in Figure 2–7) and other related document tools. Other features you can employ include Content Presenter, content data controls, and REST APIs.

Figure 2–7   The Document Explorer
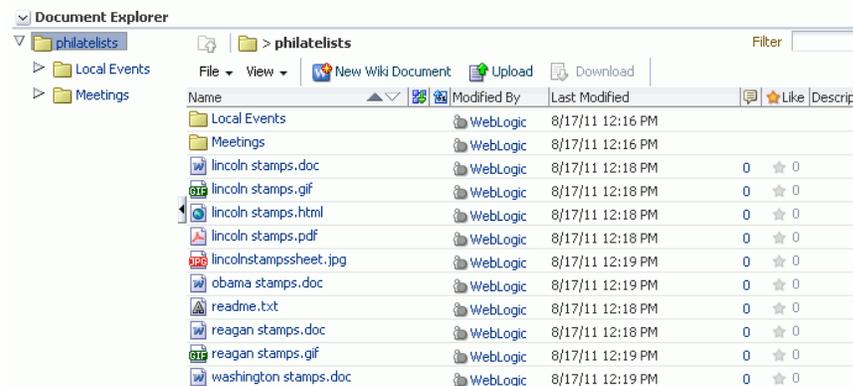
In a Portal Framework application, security must be configured between the application and the content repository (this is not a requirement if you are using Portal Builder).

For more information, see:

- "Managing Content Repositories" in *Administering Oracle WebCenter Portal*

- "Configuring Security Between Content Server and WebCenter Portal Framework Applications" in *Administering Oracle WebCenter Portal*

- "Configuring Content Repository Connections" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*

- "Introduction to Integrating and Publishing Content" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*

2.2.2.4 Administering Portal Framework Applications

Portal Framework applications can provide a WebCenter Portal Administration Console, shown in Figure 2–8. The console allows authorized users to manage application-wide preferences, manage users and roles, manage assets, manage content, create polls, register producers and external applications.

Figure 2–8   The Administration Console for Portal Framework Applications
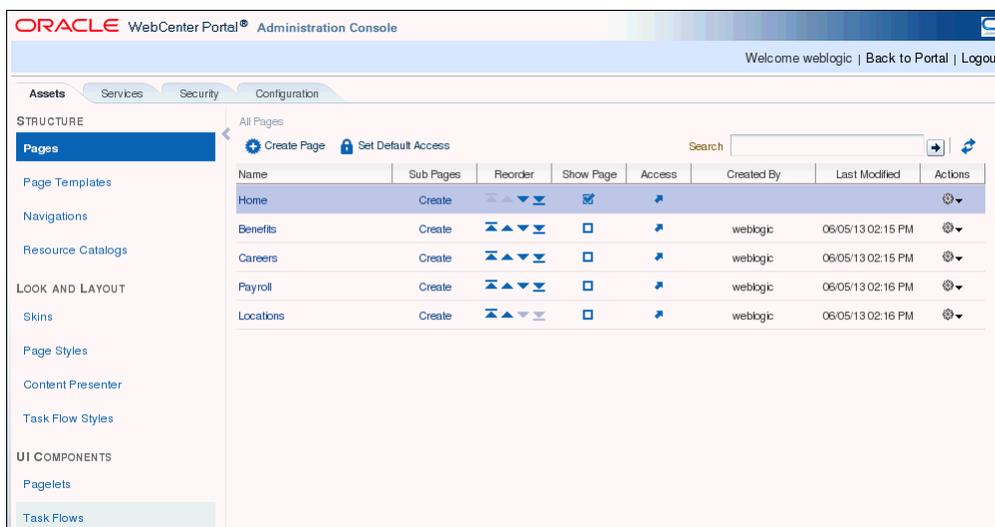
Application role assignment is the responsibility of the system administrator. Administrators can assign members a default application role or create additional, custom roles specific to their portal application.

Users with the `Administrator` role can set application-wide preferences, manage assets, manage content, create polls, and register producers and external applications.

Administrators can also manage users and roles for the application, and delegate or revoke privileges to or from other users.

For more information, see:

- "Administering Portal Framework Applications Using the Administration Console" in *Administering Oracle WebCenter Portal*.

2.2.3 Comparing Portal Builder and Portal Framework Approaches

Table 2–1 compares features and characteristics of Portal Builder and Portal Framework. Use this table to help guide your decision about which approach to choose.

> **Note:** Oracle recommends creating new portals using Portal Builder, rather than Portal Framework. While custom Portal Framework applications may provide a greater degree of flexibility when developing your portal solution, such flexibility typically forfeits product supportability and the ability to upgrade to new WebCenter Portal releases. Using Portal Builder avoids these constraints without sacrificing complexity or scalability.

Table 2–1 Comparison of Portal Builder and Portal Framework Features

Common Portal Features or Requirements	Portal Builder	Portal Framework
Packaged, pre-configured portal building solution	✓ A packaged portal-building solution that runs on a preconfigured portal server.	✗ Portal developers design and configure their own custom Oracle ADF application and manage deployment to a custom server.
Multiple portals	✓ Multiple portals can be created and managed within a single Oracle WebCenter Portal deployment.	✗ Single portal solution. Each Portal Framework application is a separate portal instance. If you want to host multiple portals on your server, a separate application must be created and deployed for each portal in JDeveloper. Hosting multiple Portal Framework applications (J2EE applications) will consume more resources.
Portal hierarchies	✓ Supports hierarchical creation of portals (subportals), a useful feature for portal organization.	✗ Does not support subportals.
Portal templates	✓ Provides several out-of-the-box portal templates. Also supports saving any portal as a portal template to create similar portals.	✗ Does not support portal templates. Developers create each portal from scratch.

Table 2–1   (Cont.) Comparison of Portal Builder and Portal Framework Features

Common Portal Features or Requirements	Portal Builder	Portal Framework
Portal tools and services	✓	✗
	Full support for Oracle WebCenter Portal tools and services.	Does not support: ■　Notes Some restrictions for: ■　Announcements (supports global system announcements only, not forum-level announcements) ■　Activity Graph ■　Discussions ■　Documents ■　Events ■　Search ■　REST See Section 2.3, "Planning for Tools and Services" for details.
Device support	✓	✓
	Portals display on desktop browsers, smart phones, and tablets out-of-the-box. Includes tooling for configuring device support, as needed.	Mobile support must be developed from scratch.
Personal portal management	✓	✗
	Includes a Home portal. The Home portal is the area where users have access to their profile, preferences, personal pages, available portals, and other features.	Does not provide a Home portal feature or any built-in features for personal pages or preferences.
Intuitive user interface	✓	✓
	Primary tooling is browser-based. The user interface is designed for ease of use.	Primary tooling is JDeveloper-based. Basic browser-based tooling for portal administrators is available out-of-the-box.
Built-in security	✓	✗
	Security features are built in to the portal.	Security features are configured from scratch by the developer.
	Security configuration between the portal and Oracle WebCenter Content is built-in.	Security must be configured between the portal and the content repository.

Table 2–1   (Cont.) Comparison of Portal Builder and Portal Framework Features

Common Portal Features or Requirements	Portal Builder	Portal Framework
	The self-registration feature makes it easy for new users to create an account. Several prebuilt workflows handle portal subscription requests and portal membership notifications.	No portal registration or subscription features are provided.
Portal administration tools	✔	✔
	Portal administration tools are packaged but not extensible.	Custom portal administration tools can be created and integrated, as needed.
	✔	✘
	Administrators can push out information relevant to specific business roles so individuals can stay on top of key updates for their line of business.	Does not support business role pages.
Portal lifecycle tools	✔	✔
	Import and export features enable moving a portal from one environment to another.	WebLogic Scripting Tool (WLST) scripts and other tools are provided to assist with portal propagation from one environment to another.
Performance dashboard	✔	✘
	In Enterprise Manager, a custom home page for Portal Builder deployments provides a series of dashboards to report status, performance and availability of all the components, tools, and services used by individual portals, as well as for the portal server.	No dashboard support. However, a wide range of page, document, portlet, and WebLogic Server metrics are available for Portal Framework applications. These metrics are in addition to the standard J2EE deployment metrics that are available through Enterprise Manager.

Table 2–1   (Cont.) Comparison of Portal Builder and Portal Framework Features

Common Portal Features or Requirements	Portal Builder	Portal Framework
Page performance analysis	✓	✗
	Includes a portal page analyzer to help diagnose slow pages, enabling system administrators to determine at a glance the components that are slowing performance.	No support for page performance analysis.
Customizable using a browser	✓	✓
	Designed for evolution in the browser. Knowledge workers and application specialists are most comfortable with this approach.	Developers may integrate browser-based customization features, if needed.
Customizable with JDeveloper	✓	✓
	Some custom components can be built in JDeveloper, including: ADF task flows, data controls, managed beans, and other assets such as skins, page styles, task flow styles, page templates, content present display templates, navigations, and resource catalogs. Requires developer expertise.	Designed for evolution in JDeveloper. Developer has complete control over the application, its configuration, and deployment. Requires developer expertise.

2.3 Planning for Tools and Services

WebCenter Portal offers many *tools and services* that allow portal members to collaborate and communicate. If WebCenter Portal is installed and configured correctly and your system administrator has set up valid connections to the required external back-end servers, tools and services are available for use in a portal.

Table 2–2 lists the tools and services that you may wish to use in a portal, and planning considerations before choosing the Portal Builder or Portal Framework approach.

Table 2–2   Planning for Oracle WebCenter Portal Tools and Services

Tools and Services	Description	Planning Considerations
Activity Graph	Recommends people, portals, and content that a user may be interested in connecting with, based on existing connections and shared interaction with objects in the portal.	**Portal Framework:** The page view feature requires a snippet of event code to be added to each page.
Activity Stream	Provides a streaming view of the activities of your connections, actions taken in portals, and business activities.	There are no differences to consider between Portal Builder and Portal Framework for activity stream.
Analytics	Provides the ability to display usage and performance metrics for your portal application.	There are no differences to consider between Portal Builder and Portal Framework for analytics.
Announcements	Provides the ability to post announcements about important activities and events to all authenticated users.	**Portal Builder:** Includes two task flows for displaying announcements: ■ Announcements ■ Announcements - Quick View Supports global system announcements and forum-level announcements. **Portal Framework:** Includes one task flow for displaying announcements: ■ Announcements Supports global system announcements only, not forum-level announcements
Discussions	Provides the ability to create threaded discussions, posting and responding to questions and searching for answers.	In general, there are no differences to consider between Portal Builder and Portal Framework for discussions. However, use of RSS in a discussion forum is only supported in Portal Builder.
Documents	Provides content management and storage capabilities, including file upload, file and folder creation and management, file check out, versioning, and so on. Exposes these capabilities through document task flows such as Document Explorer, Content Presenter, and more.	**Portal Builder:** Document services are automatically available; a folder is created on Content Server with the appropriate access permissions. **Portal Framework:** No automatic provisioning occurs, and you must configure your own access permissions. In a Portal Framework application, security must be configured between the application and content repository (this is not a requirement if you are using Portal Builder).
Events	Provides the ability to create and maintain a schedule of events relevant to a wider group of authenticated users. Also provides access to your personal events from your Outlook calendar if the Exchange server is configured.	**Portal Builder:** Supports personal events *and* portal events. Portal events are related to a specific portal and they display to all members of the portal on the portal's dedicated Events page (which can be exposed by the portal moderator) or in any events task flow that is placed on a page in the portal. Personal events come from your Microsoft Exchange calendar and display only to you. **Portal Framework:** Only supports personal events.

Table 2–2   (Cont.) Planning for Oracle WebCenter Portal Tools and Services

Tools and Services	Description	Planning Considerations
Instant Messaging and Presence	Provides the ability to observe the status of other authenticated users (online, offline, busy, or away) and to contact them instantly.	There are no differences to consider between Portal Builder and Portal Framework for instant messaging and presence.
Lists	Provides the ability to create, publish, and manage lists.	**Portal Builder:** Portal moderators can create, delete, or modify lists for their portals. They can delegate these permissions to others as needed. **Portal Framework:** By default, only administrators can create, delete, or modify lists, although this permission can be changed by the administrator when the portal is deployed.
Mail	Provides easy integration with IMAP and SMTP mail servers to enable users to perform mail functions, such as reading messages, creating messages with attachments, replying to or forwarding messages, and deleting messages.	There are no differences to consider between Portal Builder and Portal Framework for mail.
Messages and Feedback	Provides the ability to post messages, attachments, and feedback for your connections and to the activity stream.	There are no differences to consider between Portal Builder and Portal Framework for messages and feedback.
Notes	Provides the ability to "jot down" and retain bits of personally relevant information.	Only available in Portal Builder.
Notifications	Provides a means of subscribing to services and application objects and, when those objects change, receiving notification across one or more messaging channels.	There are no differences to consider between Portal Builder and Portal Framework for notifications.
People Connections	Provides the ability to manage personal profiles, access the profiles of other users, provide *ad hoc* feedback, post messages, track activities, and connect with others.	There are no differences to consider between Portal Builder and Portal Framework for people connections.
Polls	Enables you to survey your audience (such as their opinions and their experience level), check whether they can recall important information, and gather feedback.	There are no differences to consider between Portal Builder and Portal Framework for polls.
Profiles	Provides views of users' contact information (such as e-mail address, business address, phone number), department, manager, photo, portal activities, public documents, and connections.	There are no differences to consider between Portal Builder and Portal Framework for profiles.
Recent Activities	Provides a summary view of recent changes to documents, discussions, and announcements.	**Portal Builder:** The recent activities task flow shows recent activities for the specific portal in which it is used. **Portal Framework:** All recent activities are shown.

Table 2–2   (Cont.) Planning for Oracle WebCenter Portal Tools and Services

Tools and Services	Description	Planning Considerations
RSS	Provides the ability to access the content of many different web sites from a single location—a news reader.	RSS is supported in both Portal Builder and Portal Framework. However, in Portal Framework RSS is not supported within discussion forums.
Search	Provides search functionality. This includes integrating Oracle Secure Enterprise Search (SES).	**Portal Builder:** Page metadata, portal metadata, profile, and list are included for search, in addition to documents, discussions, and announcements. **Portal Framework:** By default, search uses crawl-based search, but SES configuration must be performed. Also, only documents, discussions, and announcements are included in a search.
Tags	Provides the ability to assign one or more personally-relevant keywords to a given page or document	**Portal Builder:** Users can tag pages and documents. **Portal Framework:** Allows tagging custom objects, in addition to pages and documents.
Worklists	Provides access to all Business Process Execution Language (BPEL) worklist items assigned to the user. Worklist notifications and alerts are queried from all of the BPEL servers configured in WebCenter Portal.	There are no differences to consider between Portal Builder and Portal Framework for worklists.

For more information about tools and services, see:

- "Introduction to Portal Tools and Services" chapter in *Building Portals with Oracle WebCenter Portal*

- "Preparing Your Application for WebCenter Portal Tools and Services" chapter in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*

2.4 Planning for a Consistent Look and Feel

An important portal planning activity involves considering the layout, style, and organization of your portal and delegating their actual implementation. It is common for an enterprise portal to include hundreds or even thousands of pages, and planning for a consistent look and feel across all pages is a crucial step.

In Oracle WebCenter Portal, templates, page styles, skins, navigations, and other assets are available for ensuring consistency across portal pages. When you create a new portal or a new page, picking from a catalog of these assets makes it easy to achieve a consistent user experience and an appropriate mix of features.

Table 2–3 briefly describes features related to look and feel and provides links to implementation details in the Oracle WebCenter Portal documentation. The table lists the look and feel assets that help you achieve consistency across the portal; however, in practice, the first assets you will probably choose to implement are page templates, page styles, skins, and navigation components.

> **Note:** Sample implementations of many of the templates and other assets that are used to achieve consistency are provided with Oracle WebCenter Portal out-of-the-box. These seeded components can be used directly or as a starting point for creating new custom components.

*Table 2–3   Portal Look and Feel Related Assets*

Look and Feel Components	Description	Portal Builder	Portal Framework
Portal Template	Portal templates provide an easy way to get started creating a portal. You can save any existing portal as a portal template. If a tool, such as documents, is enabled in the template, it will be enabled in the portal created from that template. See the "Working with Portal Templates" chapter in *Building Portals with Oracle WebCenter Portal*.	✓	✗
Page Template	Page templates control the layout and common content of portal pages across the entire portal. Page templates typically define header, footer, and content regions for a page. In addition, they can be used to add brand-specific elements, like logos, navigational elements, and action elements such as login and logout buttons. For using page templates in Portal Builder, see the "Working with Page Templates" chapter in *Building Portals with Oracle WebCenter Portal*. For developing page templates from scratch, or modifying existing page templates in JDeveloper, see the "Developing Page Templates" chapter in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.	✓	✓
Page Style	Page styles also control aspects of the layout of a page, but focus more on the way content is arranged in the page. Page styles can be used, for example, to ensure the consistent arrangement of content and the types of content that appear in pages. For the out-of-the-box page styles provided with Portal Builder, see the "Working with Page Styles" chapter in *Building Portals with Oracle WebCenter Portal*. For developing page styles from scratch, or modifying existing page styles in JDeveloper, see the "Developing Page Styles and Task Flow Styles" chapter in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.	✓	✓
Skins	CSS-based skins work together with page templates to determine the overall look and feel of the pages in your portal. While the page template controls the location and behavior of components on the page, the skin controls the appearance of those components. For the out-of-the-box skins provided with Portal Builder, see the "Working with Skins" chapter in *Building Portals with Oracle WebCenter Portal*. For developing skins from scratch, or modifying existing skins in JDeveloper, see the "Developing Skins" chapter in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.	✓	✓

Table 2–3 (Cont.) Portal Look and Feel Related Assets

Look and Feel Components	Description	Portal Builder	Portal Framework
Navigations	Navigation models have a structure (like tabs, breadcrumbs, or tree), and specify the specific targets to navigate to.	✓	✓
	For Portal Builder, see the "Working with Navigation Models" chapter in *Building Portals with Oracle WebCenter Portal*.		
	For Portal Framework, see the "Developing a Navigation Model" chapter in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.		
Display Templates	Display templates define how Content Presenter displays content items (including images and text) in a portal. Content Presenter enables you to precisely customize the selection and presentation of content in a portal.	✓	✓
	Note: New display templates must be created in JDeveloper.		
	For Portal Builder, see the "Publishing Content Using Content Presenter" chapter in *Building Portals with Oracle WebCenter Portal*.		
	For Portal Framework, see the "Creating Content Presenter Display Templates" chapter in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.		
Task Flow Styles	Task flow styles are templates used for creating task flows. They ensure consistency in the way task flows are rendered.	✓	✓
	For Portal Builder, see the "Working with Task Flow Styles" chapter in *Building Portals with Oracle WebCenter Portal*.		
	For Portal Framework, see the "Developing Task Flow Styles" section in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.		

> **Note:** In Portal Builder, an administrator can set a default page template for all portals. A portal moderator (a user with manage permissions in the portal) can override the global default and set a default page template for a specific portal. The default cannot be overridden on specific pages. In this way, layout consistency is ensured across the entire portal. Figure 2–9 shows the administration user interface for setting the default page template in Portal Builder, where the default page template is selected from a list.

Figure 2–9   Selecting a Default Page Template in Portal Builder
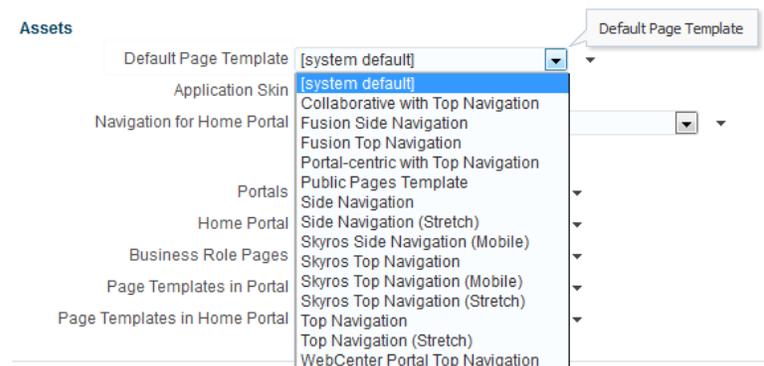
2.5 Planning for Portal Security

Typically, users must log in to the portal and, based on role-based delegated security settings made by an administrator, they see only the information and resources that they are authorized to see. In other words, what you see in a portal depends on the role you have in the portal. For example:

- Employees log in and securely change their personal information, such as address or phone number.

- Human Resources personnel add, remove, or update content in the specific parts of the portal that they have permission to edit.

- Managers view information about their direct reports, such as salary history and performance reviews.

- Business partners access only their own orders. In this case, the portal may pass the security context to the underlying application through single sign-on. Orders may be viewable only within a certain time or the actions available are based on region or partner status.

Oracle WebCenter Portal handles all of these diverse aspects of security.

> **Note:** When you create a new portal in Portal Builder, you can set overall access to be public, private, or hidden. Anyone can visit a public portal. Private and hidden portals require some kind of authorization, such as membership required through an invitation. Hidden portals are not shown in the list of available portals and do not appear in search results.

Through standard security mechanisms, such as single sign-on, authentication credentials can be passed to the underlying applications, giving the user a one-login experience when interacting with application data through the portal.

Access to portal pages and components can be further refined through delegated administration. Delegated administration provides a mechanism for securing portal resources based on user roles. For example, you can allow users in one role (managers, for instance) to access all portal features, but deny certain features to users in another role (employees, for instance).

Delegated security is particularly important for portals with many pages. Because security settings are inherited from parent pages to child pages (within the page hierarchy), it is not necessary to set security for each individual page.

For more information about security for Portal Builder, see:

- "Managing Security Across Portals" in *Administering Oracle WebCenter Portal*.

- "Administering Security in a Portal" in *Building Portals with Oracle WebCenter Portal*.

- "Granting Users Access to a Portal" in *Building Portals with Oracle WebCenter Portal*.

For more information about security for Portal Framework applications, see:

- "Securing Your WebCenter Portal Framework Application" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*.

- "Default Security Configuration" in *Administering Oracle WebCenter Portal*.

- "Configuring Portal Framework Applications for Single Sign-on" in *Administering Oracle WebCenter Portal*.

■ "Managing Oracle WebCenter Portal Security" in *Administering Oracle WebCenter Portal*.

2.6 Post-Planning Tasks Summary

Table 2–4 lists tasks you will perform after completing the portal planning stage, with reference to more details for the development approach you select.

Note: Oracle recommends creating new portals using Portal Builder, rather than Portal Framework. While custom Portal Framework applications may provide a greater degree of flexibility when developing your portal solution, such flexibility typically forfeits product supportability and the ability to upgrade to new WebCenter Portal releases. Using Portal Builder avoids these constraints without sacrificing complexity or scalability.

Table 2–4 Common Portal Setup and Development Tasks

Task	Portal Builder Details	Portal Framework Details
Install software and configure global settings for Oracle WebCenter Portal.	See Section 1.5, "Navigating the Oracle WebCenter Portal Documentation Set" for installation and administration guides.	See Section 1.5, "Navigating the Oracle WebCenter Portal Documentation Set" for installation and administration guides. In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Setting Up Your Development Environment"
Determine the type of portal to build based on your requirements.	See Section 1.1, "What Is a Portal?"	See Section 1.1, "What Is a Portal?"
Consider the portal life cycle. Plan for development, testing, staging, and production environments.	See Section 2.2.1.1, "Understanding the Portal Life Cycle with Portal Builder"	In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Working Productively in Teams" ■ "Understanding the WebCenter Portal Framework Application Life Cycle" ■ "Deploying and Testing Your Portal Framework Application"

Table 2–4   (Cont.) Common Portal Setup and Development Tasks

Task	Portal Builder Details	Portal Framework Details
Create back-end connections to content repositories, services, and databases, as needed.	See Section 2.2.1.3, "Integrating Content into a Portal with Portal Builder" In *Administering Oracle WebCenter Portal*, see: ■ "Managing Content Repositories" ■ "Managing Tools and Services"	In *Administering Oracle WebCenter Portal*, see: ■ "Managing Content Repositories" ■ "Configuring Security Between Content Server and WebCenter Portal Framework Applications" ■ "Managing Tools and Services" In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Configuring Content Repository Connections" ■ "Introduction to Integrating and Publishing Content" ■ "Configuring WebCenter Back-End Services"
Develop a security plan. Add delegated security to pages, assets, and membership to the portal	See Section 2.5, "Planning for Portal Security"	See Section 2.5, "Planning for Portal Security"
Choose the services you wish to add to the portal. For example, the documents service allows documents to be stored, archived, shared, and edited. The activity service allows the activities of portal members to be tracked.	See Section 2.3, "Planning for Tools and Services" In *Building Portals with Oracle WebCenter Portal*, see: ■ "Introduction to Portal Tools and Services"	See Section 2.3, "Planning for Tools and Services" In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Preparing Your Application for WebCenter Portal Tools and Services" in *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*
Create a new portal.	In *Building Portals with Oracle WebCenter Portal*, see: ■ "Creating and Building a New Portal"	In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Creating WebCenter Portal Framework Applications"

Table 2–4   (Cont.) Common Portal Setup and Development Tasks

Task	Portal Builder Details	Portal Framework Details
Build up the portal structure by adding pages, portal assets, navigation elements, and other features to it.	In *Building Portals with Oracle WebCenter Portal*, see: ■ "Creating and Editing a Portal Page" ■ "Creating, Editing, and Managing Portal Assets"	In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Introduction to Portal Resource Management"
Add content to the portal (requires a connection to a content repository, such as Oracle WebCenter Content). ■ Use document tools to view and manage documents and other types of content in your organization's content repositories. ■ Consider using Content Presenter to include targeted content.	See Section 2.2.1.3, "Integrating Content into a Portal with Portal Builder" In *Building Portals with Oracle WebCenter Portal*, see: ■ "Introduction to Working with Content" ■ "Publishing Content Using Content Presenter" ■ "Working with Document Task Flows and Document Components"	See Section 2.2.2.3, "Integrating Content into a Portal with Portal Framework" In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Introduction to Integrating and Publishing Content" ■ "Configuring Content Repository Connections"
Make iterative changes to the portal as needed. Redeploy or move changes between environments.	See Section 2.2.1.1, "Understanding the Portal Life Cycle with Portal Builder" In *Administering Oracle WebCenter Portal*, see: ■ "Understanding WebCenter Portal Life Cycle" In *Building Portals with Oracle WebCenter Portal*, see: ■ "Publishing a Portal"	Section 2.2.2.1, "Understanding the Portal Life Cycle with Portal Framework" In *Developing Portals with Oracle WebCenter Portal and Oracle JDeveloper*, see: ■ "Understanding the WebCenter Portal Framework Application Life Cycle" In *Administering Oracle WebCenter Portal*, see: ■ "Deploying Portal Framework Applications"