IBM

IBM® WebSphere® Portal Server Product Architecture V2.1

November 19, 2001

Abstract

Portals provide a secure, single point of access to diverse information and applications, personalized to the needs of their users. WebSphere Portal Server provides an open and extensible framework plus the flexible and scalable infrastructure needed for many types of portals supporting B2C, B2B, and B2E usage models. WebSphere Portal Server provides the base on which to build enterprise, marketplace, consumer, and workspace portals accessible from a wide variety of desktop and mobile devices.

This technical white paper gives an in-depth view of the architecture and design of WebSphere Portal Server, including its presentation and portlet framework, security, user management, personalization, content management, performance and scalability, search, and enterprise information connectivity features. Also, because WebSphere Portal Server Version 2.1 is packaged within the WebSphere Portal Family solutions, information about how WebSphere Portal Server leverages the components included with the WebSphere Portal Family solutions is provided.

Table of contents

Introduction

A *portal* provides a single point of access to diverse information and applications, secure interactions, a customizable interface, personalized content, and much more.

Many portals exist, including enterprise information portals, business-to-business marketplaces, employee workspaces, and public Web portals. No matter what type of portal is in use, the general requirements for all portals are the same. All portals require a scalable infrastructure that can change with business expansion, a flexible and powerful presentation framework that produces an aesthetic interface, and a framework on which to build portal components easily. In addition, most portals require personalization, which enables a portal to deliver relevant information to the user. Personalization provides a more productive and interactive experience for the user, as well as promoting user loyalty to the portal.

Some portals have unique requirements. If sensitive information passes through the portal, the portal might employ specialized forms of authentication and access control to heighten security If a portal requires high availability, such as a consumer portal that is available over the Internet, the portal might require user self-registration and the flexibility for users to manage their own accounts within a large user database. Finally, if an existing user database is within an enterprise, the portal must integrate with that user database and possibly take advantage of existing enrollment systems.

WebSphere Portal Server offers a framework to meet all the presentational, security, scalability, and availability issues described above. Also, WebSphere Portal Server provides the open and flexible infrastructure for creating and deploying many types of portals that are accessible from a wide variety of desktop and mobile devices.

The architecture -- an illustration

Figure 1 provides an illustration of the overall architecture for WebSphere Portal Server. Use the illustration as a guide for the sections provided in this paper, including the presentation services, portal infrastructure, and portal services.

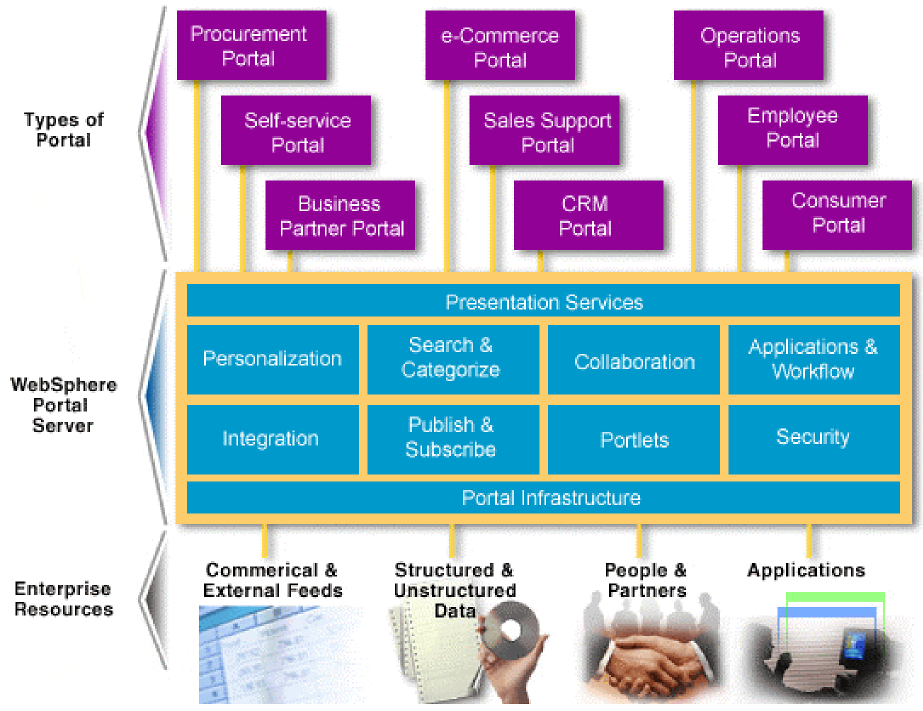


Figure 1. Overall architecture for WebSphere Portal Server

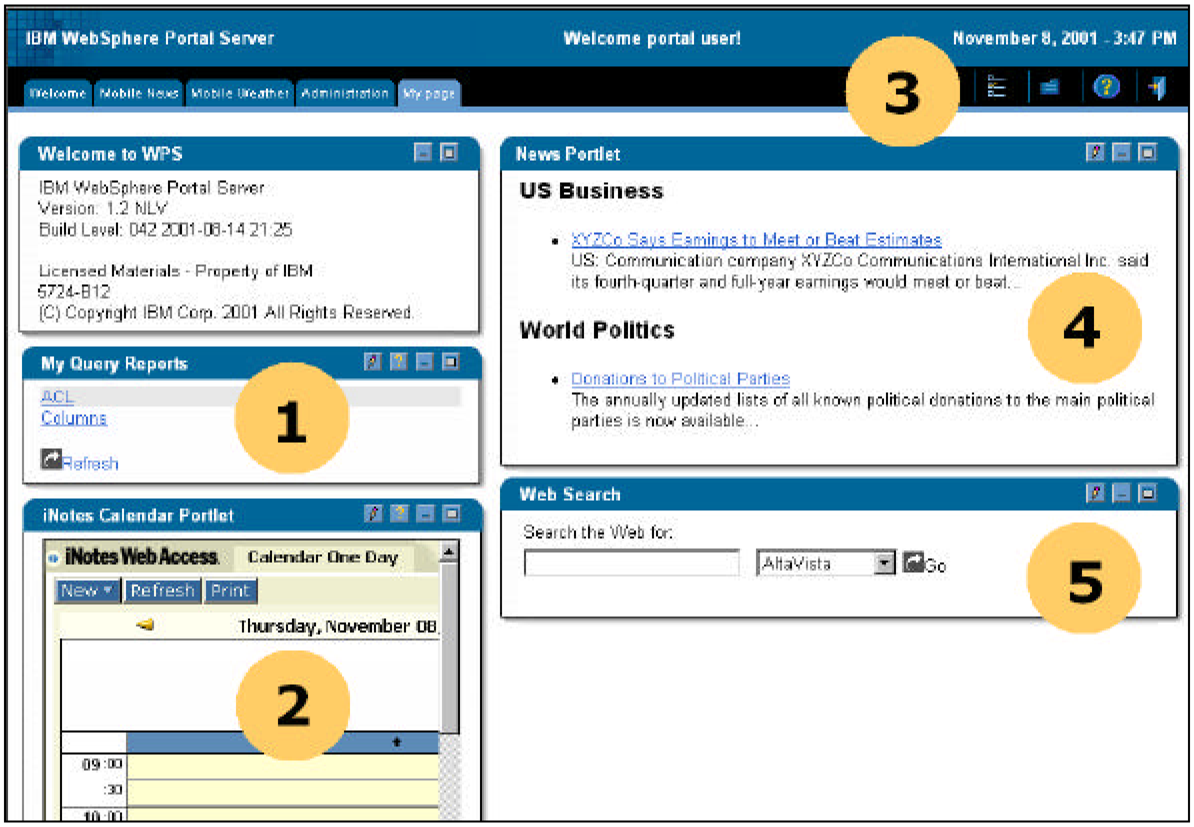
The portal framework simplifies development and maintenance.

- The page structure is defined only once.
- Portlets are defined independently.
- Portlets can be changed without impacting the overall page design.
- Targeting multiple browsers and mobile devices is made easier.

Presentation services

The WebSphere Portal Server framework produces a customized and personalized home page for users in which the content for the page is aggregated from a variety of content and application data sources. The content areas or *portlets* display according to what is available and how users customize their portal pages.

Figure 2 shows a portal page as rendered by a desktop Web browser, which is commonly used to access a portal. Applications such as browsers for wireless phones also provide portal access. Although every portal is unique, typical portal features are shown in figure 2 and described in subsequent text.



Welcome to WPS

IBM WebSphere Portal Server
Version: 1.2 NLV
Build Level: 042 2001-08-14 21:25

Licensed Materials - Property of IBM
5724-B12
(C) Copyright IBM Corp. 2001 All Rights Reserved.

My Query Reports

ACL
Columns

Refresh

iNotes Calendar Portlet

iNotes Web Access Calendar One Day
New Refresh Print
Thursday, November 08
09:00
:30
10:00

News Portlet

US Business

- XYZCo Says Earnings to Meet or Beat Estimates
  US: Communication company XYZCo Communications International Inc. said its fourth-quarter and full-year earnings would meet or beat...

World Politics

- Donations to Political Parties
  The annually updated lists of all known political donations to the main political parties is now available...

Web Search

Search the Web for: AltaVista Go

Figure 2. Portal home page

1. Custom applications for business intelligence and enterprise-resource planning deliver timely and relevant information.
2. Collaboration applications keep users in touch with colleagues or customers in real-time and centralize access to e-mail, calendar, address books, and to-do list.
3. The customize link lets users adjust their profile and modify their home page contents.

4. The News portlet shows the latest news headlines from a syndicated content provider such as Reuters, Dow Jones, Newswire, or Business Wire.
5. The Search portlet gives quick access to both Internet content and local documents for searching.

The portal engine

WebSphere Portal Server provides a pure Java portal engine, which runs on multiple hardware platforms and operating systems. The main responsibility of the portal engine is to aggregate content from different sources and to serve the assembled content to multiple devices. Additionally, the portal engine decouples the presentation details of the portal page from the characteristics of the portlets. This separation enables each portlet to be developed and maintained as a discrete component, which in turn enables faster, easier, and specialized development for the overall portal site.

Figure 3 provides an illustration of the portal engine components. Subsequent explanatory notes are provided.

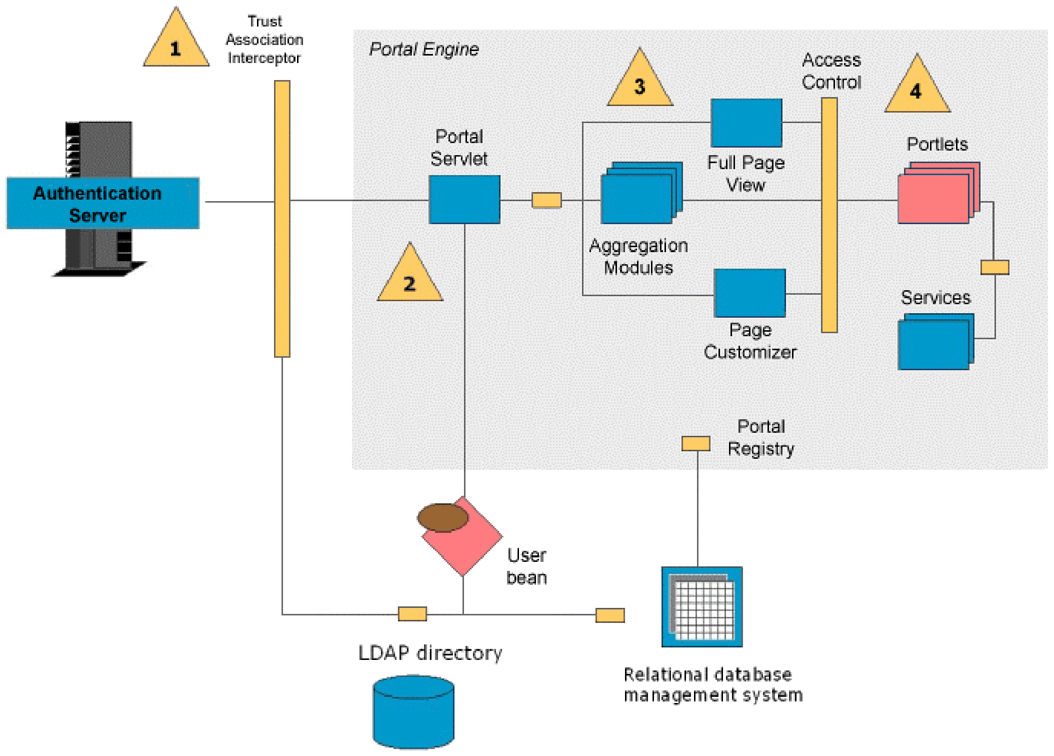


Figure 3. Portal-engine components

1. In front of the portal engine is an authentication component such as standard WebSphere security, WebSEAL-Lite, or a third-party authentication proxy server.
2. The central component in the portal engine is the *portal servlet*. It examines the URL and header fields of each request and invokes the appropriate handler. The portal servlet handles the request in two

phases. In the first phase, portlets have an opportunity to send event messages to other portlets. For example, portlets might send events in order to update data that will be rendered in the next phase. In the second phase, the appropriate aggregation module for the user device renders multiple portlets in a single page. The aggregation modules accumulate information from each portlet, add standard decorations around the portlet (such as a title bar, edit button, and enlarge button), position it on the page, and generate the overall page markup.

3. **Access to portlets is controlled by checking access rights during page aggregation, page customization, and other access points such as viewing the portlet in its maximized state.**

Page content aggregation

Currently, WebSphere Portal Server has three aggregation modules. The HTML aggregation component produces pages for desktop computers and other devices with HTML browsers. The Wireless Markup Language (WML) aggregation component produces WML content for Wireless Access Protocol (WAP) devices, such as mobile phones. The iMode aggregation component produces cHTML markup for mobile devices in the NTT DoCoMo network. In future releases, the portal will provide additional aggregation components, including a voice aggregation module for devices with VoiceXML browsers and a Personal Digital Assistant (PDA) aggregation module for PDA devices.

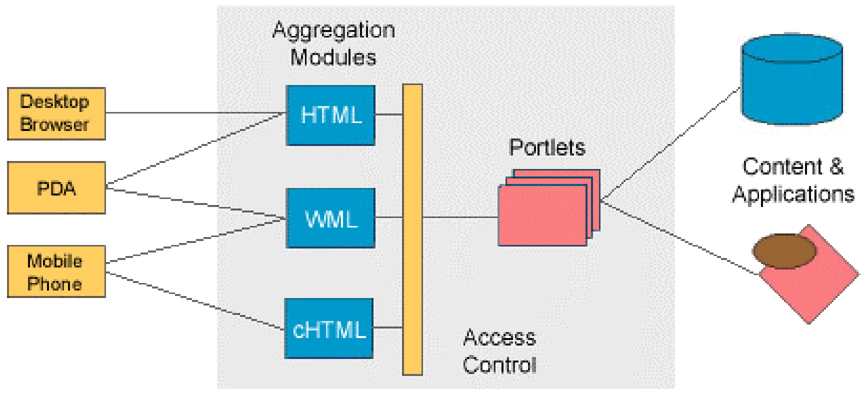


Figure 4. Page content aggregation

Each user can customize a unique home page for each device, selecting the content and applications that are most useful on the device. When the home page is requested, page aggregation works by first detecting the type of device that is making the request, and then assembling the portlets which render their contents in the appropriate markup language.

Multi-device portlets

When a user customizes the home page for a particular device, the portlet selection list only shows portlets that can actually produce markup appropriate for that device. Thus, the list of available portlets for each device depends on what the portlets can actually do.

Some portlets may be available for all the supported devices, while others may be available only on a single device. The user interface design of each portlet also varies from device to device. Thus, the user home page and each of the portlets might be very different on a mobile phone.

In general, WebSphere Portal Server produces platform and browser-neutral markup, which works with many Web browsers, including Microsoft Internet Explorer 4.0 or later, as well as Netscape Navigator 4.7 or later. Some mobile phone browsers, and PDA devices running HTML or WML browsers also work. Most browsers that support HTML 3.2, WML 1.1 or 1.2, and iMode 1.0 will work, though specific devices should always be verified. Depending on the needs of a particular portal or set of portlets, it is equally possible and valid for portlets to generate markup that is targeted to Web browsers that support later HTML standards, such as Internet Explorer 5 or Navigator 6.

The portal home page

The portal home page is the main page which contains the attributes for the presentation attributes of the portal. The structure of the page and its navigation areas is defined in page template files, using JSP markup. The arrangement and the look of these areas can be changed easily so that the portal pages reflect a style or brand image.

Navigation areas can be placed anywhere, such as a masthead area at the top of the page or on the left-hand side. Cascading style sheets, images, and other visual elements are used to further define the look of the page. Using these techniques, the contents of the navigation areas are determined by the portal site developer. The precise details differ for each mark-up. For example, WML decks are also structured using JSP templates, but they do not use all the same graphic elements that HTML pages have.

When building the body of the page where the portlets are displayed, WebSphere Portal Server uses page layout templates, which are also based on JSP pages. The process to aggregate a page works as follows:

1. **When the aggregation module is invoked, it obtains the page layout information for the current page of the current user.**
2. **The page layout information is parsed and generates a tree with nodes referencing rows, columns, portlet decoration elements, and finally the actual portlets.**
3. **As the tree is traversed, JSP templates corresponding to each node are included in the output stream. Each node representing a row or column includes the appropriate row or column template. Each node that represents a portlet decoration includes another JSP template. For the portlet nodes, the actual markup for that portlet is included.**

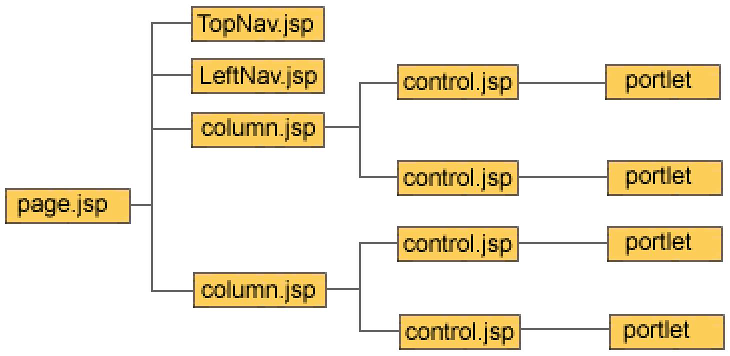


Figure 5. JSP-based aggregation

Just as the page layout and navigation areas can be changed easily, so can the JSP row, column, and decoration templates. This makes it easy to change the look-and-feel of the portal.

Globalization and accessibility

WebSphere Portal Server loads page templates and portlets based on the browser settings or based on the preferred language specified in the user's profile. The page mark-up produced by WebSphere Portal Server meets U.S. federal guidelines and Web Accessibility Initiative guidelines for ensuring accessibility to people with disabilities.

End-user page customization

The portal engine customizer component allows users to modify the content and layout of their portal pages. End users can define one or more home page tabs, and then decide how the portlets are arranged inside the portlet display area of each tab. There are several predefined column styles, and portlets can be placed or rearranged in each column.

Portlets

Portlets are the visible components that users see on their portal pages. Portlets can be as simple as an e-mail inbox or as versatile as a sales forecast from an ERP application. From a technical point of view portlets are very similar to Java servlets, except that they only return a subset of the output page.

Portlets are the way that software vendor products or in-house custom applications can snap into the portal framework. They can be written in a variety of ways. The simplest portlets might use static HTML or WML mark-up, or perhaps JSP syntax. Intermediate portlets could use Java beans or servlets, or perhaps XML/XSL transformations. More complex portlets involve writing custom Java code.

IBM provides portlets as well as by third-party software vendors and business partners. Standard portlets provided by IBM include portlets for personal productivity applications, such as Lotus Notes® e-mail and calendar, plus many application enabling and utility portlets. You can download these portlets from the Portlet Catalog on the IBM Web site.

Standard portlets

WebSphere Portal Server provides a rich set of standard portlets which include the following:

- **Rich Site Summary (RSS) portlet: formats RSS data, commonly used for news feeds**
- **XSL portlet: transforms any XML using XSLT**
- **JSP portlet: renders any JSP file or servlet**
- **HTML and WML portlet: renders any URL**
- **Notes® portlets: includes portlets that access Lotus Notes e-mail, calendar, address book and to-do list.**
- **Exchange portlets: includes portlets that access Microsoft Exchange e-mail, calendar, address book and to-do list.**
- **Lotus Sametime™ portlet: gives access to the Sametime® instant messaging server**
- **Lotus QuickPlace™ portlet: gives access to a QuickPlace™ team room**
- **Syndicated Content portlets: provide news and other information from ScreamingMedia, YellowBrix, Factiva and Hoovers.**

Portlets can be structured to inherit configuration settings from other portlets. In this way, many new portlets can be created without writing any code. For example, a general news portlet might be defined to work with any RSS data feed. Two specific instances of the portlet might point to CNN and BBC news.

Portlet API

For cases where custom coding is required, WebSphere Portal Server includes an open standard Java API, called the *portlet API*. **The portlet API provides a stable, high performance, scalable interface for portlet writers, and it is independent of the portal engine to allow interoperability of portlets among future portal engines. The portlet API is supplemented by plug-in services, giving vendors the ability to provide value-add functions without requiring API changes from release to release of the portal server.**

The portlet API is very similar to the Java servlet API. The *Portlet* **class corresponds to the** *Servlet* **class, with method signatures matching the** *init*, *service*, **and other key methods. Similarly, the** *PortletConfig* **class corresponds to** *ServletConfig, PortletContext* **corresponds to** *ServletContext*, *PortletRequest* **corresponds to** *ServletRequest*, **and so on.**

Programmers who already know the servlet API will be very comfortable with writing portlets. Best practices for writing portlets are similar to those for writing servlets, since they have similar operational characteristics. Portlets, like servlets, are singletons; meaning that there is only one instance of each portlet class shared by all requesters. This means that instance variables and class variables in portlets should not be used. Instead, store the user information in the user session. It is important to make the portlet code as fast as possible to minimize its impact on the overall page performance.

Services, events, and access control

The portlet API allows for the definition and registration of services implementing particular interfaces. This enables a stable API core that can be extended by services as required. Standard classes provided by WebSphere Portal Server include the *User* **class (for getting user data like name, address, etc.) and the** *PersistenceService* **(for storing per-user and**

per-portlet settings). These services in turn, call the User bean interfaces that will be described later in this paper.

To enable portlet-to-portlet communication, the PortletContext object has a send method. This method allows a portlet to send a message to another portlet through the framework. The target portlet will then receive a message event and can then retrieve the message. In addition to sending messages, portlets also can share data through dynamic attributes attached to their context object.

The home page customizer, aggregation modules and views invoke portlets via the *portlet container*. **The portlet container uses the** *access control interface* **to determine whether the current user is permitted to access the portlet.**

Portlet development

WebSphere Portal Server includes several example portlets with source code. You can use these examples to learn about portlet programming techniques or as a starting point for further portlet development. Many of the example portlets can be used as a basis for writing your own portlets.

Portlets include both visual elements and processing logic. A typical portal could include class files, Web pages, images, and other related assets. All of these assets are packaged together into a jar file format, called a Portlet ARchive (PAR) file.

When writing custom portlets, a model-view-controller design is recommended, as shown in the following figure.

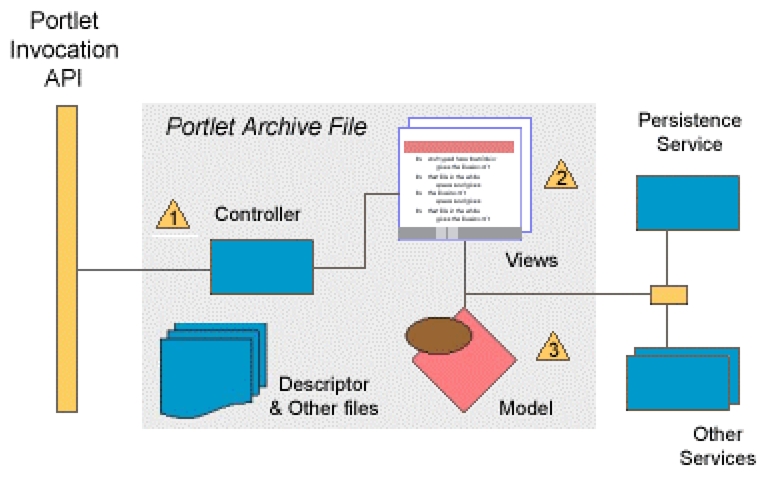


Figure 6. Portlet structure

1. **The** *controller* **is the class responsible for rendering the portlet by calling upon the appropriate view.**

2. *Views* are usually implemented as JSP pages. Portlets may have several different views, including their standard view (which renders the portlet on the home page), a maximized view (which renders the portlet in its maximized state), and an edit view (which displays a page for changing the portlet settings).
3. *Models* are data bean classes, which hold the internal settings for the portlet. This data bean also supplies data to the edit view.

To make the job easier, WebSphere Portal Server provides a generic *MVCPortlet* class that sets up the structure for you. It accepts parameters for the JSP views and the data beans. This enables simple portlets to be implemented just by creating the JSP files and beans.

Tools for portlet development

Open development platform

IBM WebSphere Studio Workbench is based on the Eclipse open development platform, which can provide a single, unified experience for portlet programming and testing.

Both IBM and third party Web application development tools can be used in portlet development. IBM offers VisualAge® for Java, which is useful as a programming and test environment for portlets. IBM WebSphere Studio is useful for creating and publishing JSP, images, and other portal assets. The WebSphere Everyplace® Server SDK includes additional tools for developing and testing mobile content and applications.

There are several steps involved in creating and deploying a portlet:

1. Implement the portlet classes using VisualAge for Java
2. Implement JSP using WebSphere Studio
3. Develop other portlet resources, such as images, data connections, multi-media content, etc.
4. Package the classes, content, and JSP files in a PAR file.
5. Add the portlet to the portal server.
6. Test the portlet by adding it to a user home page.

The PAR file is very similar to a Web application archive file. It contains an XML descriptor for configuring the portlet, plus all the related file assets such as images and JSP files. When a PAR file is deployed, it copies all file assets into the appropriate folder and adds the portlet configuration information to the portal registry.

Portal services

Portal services are features that WebSphere Portal Server leverages to provide a complete portal solution. Features such as personalization, search, content management, site analysis, enterprise application integration, collaboration, Web services, and the components of the WebSphere Portal Family solutions are discussed in this section.

Personalization

Attract and retain visitors

Personalization is an important functionality for all kinds of portals. It is a mechanism for attracting and retaining users.

Personalized content makes users more productive when visiting the portal by making the most relevant information readily available. Personalized interaction distinguishes a portal from an ordinary Web site.

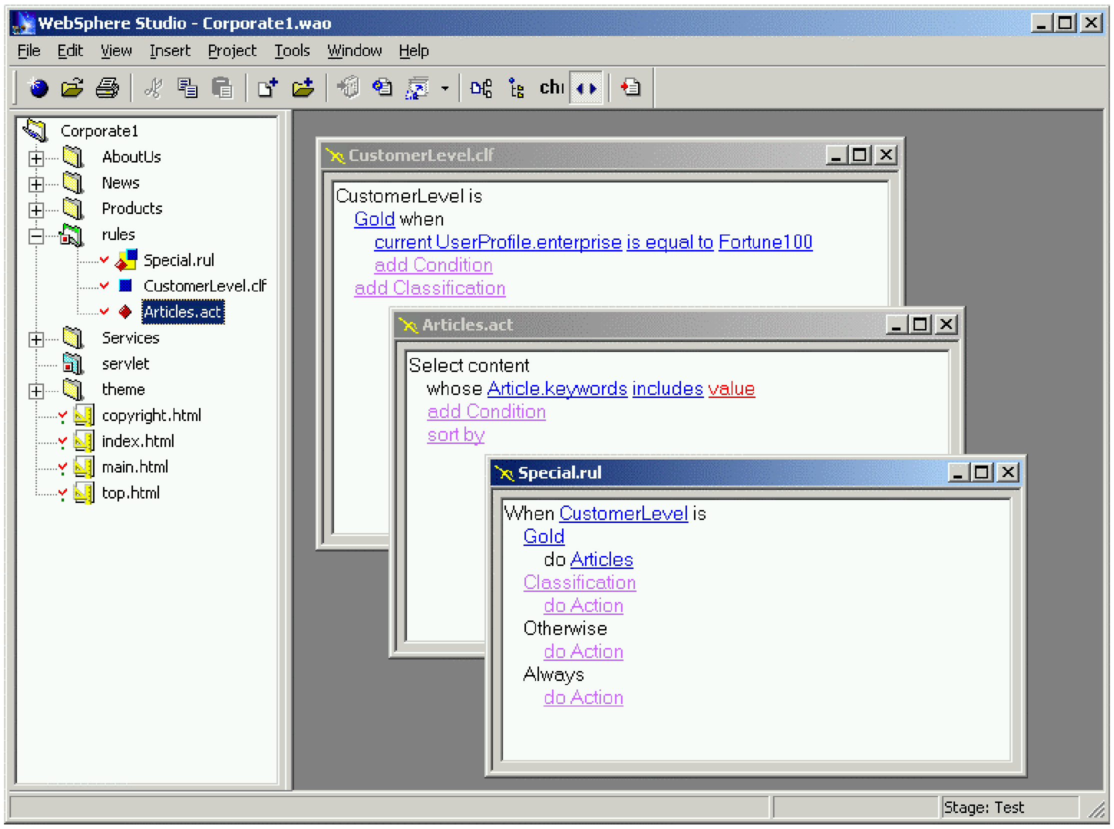
WebSphere Portal Server allows users to customize the appearance of their portal pages according to personal preferences. The customization is accomplished partly through administrative setup, which defines the default settings and access rights to portlets. Further customization is accomplished through explicit user actions to change the contents and layout of the portal home page

IBM WebSphere Personalization is integrated into and included with WebSphere Portal Server, so that advanced levels of personalization can be achieved. For example, personalization can be based on business rules and user-profile information, in addition to explicit user preferences. WebSphere Personalization goes beyond simple home page customization, and supports targeting information to specific users. It offers two advanced kinds of personalization techniques:

- The rules engine uses business logic to select content for the user. For example, a rule might display special discounts to gold customers, but only during the summer months.
- The recommendation engine uses collaborative filtering technology to select content based on common interests or behaviors. It observes click streams that can subsequently be examined for trends. This technique is often used in commerce portals for cross-selling products.

The portal server, the rules engine and the recommendation engine share user profile and content repositories. In other words, the User bean class of the portal server is already enabled for use in WebSphere Personalization Rules. Additionally, content can be stored in any data repository and is accessed through classes that implement the WebSphere Personalization Resource interface methods.

The JSP views of a portlet can use WebSphere Personalization rules and recommendations in the same way that any JSP page does. This allows the content within the portlet to be personalized, based on the rules and recommendations. Rule and recommendations can also be used in the layout JSP templates or in the page customizer JSP to provide more advanced personalization of the portal.



```
CustomerLevel is
  Gold when
    current UserProfile.enterprise is equal to Fortune100
    add Condition
  add Classification
```

```
Select content
  whose Article.keywords includes value
  add Condition
  sort by
```

```
When CustomerLevel is
  Gold
    do Articles
  Classification
    do Action
  Otherwise
    do Action
  Always
    do Action
```

Figure 7. Rule editor for WebSphere Personalization

Customers who have already used WebSphere Personalization to create personalized Web page can take advantage of the JSP portlet to reuse their JSP pages. The content model and the user profile are also reusable. If there is not already an existing user profile resource class, the one supplied with the portal server is already enabled for use with WebSphere Personalization. If there is an existing user profile resource class, then using the same LDAP repository for both the personalization and portal servers ensures that the user identity information is consistent.

Content management

WebSphere Portal Server works with content management tools such as IBM Content Manager, which is provided with some WebSphere portal solutions, and Enterprise Information Portal (EIP) Client Kit for Content Manager which offers a common client base for Content Manager. Content Manager is a portfolio of integrated software products for Web-enabled content, and EIP Client Kit for Content Manager provides a single point of access to the information assets that are stored across IBM Content Manager repositories, including all types of documents, rich media objects, computer-generated output, or collaborative PC files.

Other third-party products, such as Interwoven TeamSite, Vignette Content Management Server, FatWire UpdateEngine 5, or Documentum 4i WCM are supported. For example, the Interwoven products work with WebSphere Portal Server to help customers develop presentation templates and portlets

and also display and deploy content from the content management repository to the portal.

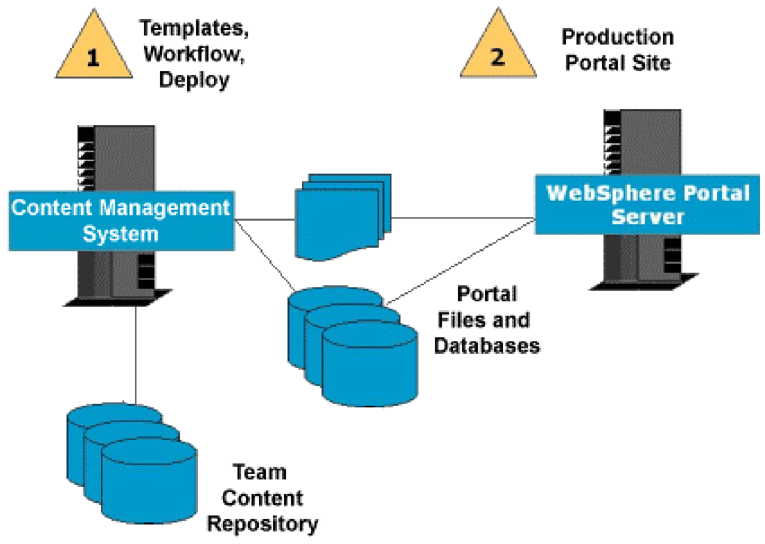

1
Templates, Workflow, Deploy
2
Production Portal Site
Content Management System
WebSphere Portal Server
Portal Files and Databases
Team Content Repository

Figure 8. Content management

Content publication and subscriptions

Content sources for portlets

News and financial content can easily be displayed through portlets that have been developed by IBM's content partners. These companies offer trial content feeds and various content offerings that you can purchase for use in your portal.

WebSphere Portal Server provides the infrastructure to facilitate connections to virtually any content or application source. The portal server provides built-in support for many common content standards, including RSS, Open Content Syndication (OCS), News Markup Language (NewsML), News Industry Text Format (NITF), and most XML or browser markup.

Content suppliers

Content suppliers offer a large variety of content for portals. Some providers offer live data through a URL (for example, Moreover and XMLTree), and others require client software that stores content locally, either as files or in a database (for example, ScreamingMedia and YellowBrix). Pricing, subscription models, and content formats vary widely.

- **Factiva, a Dow Jones & Reuters company, provides world-class global news and business information, offering access to up to 7000 highly respected global, multi-language sources such as The Wall Street Journal, The New York Times, Le Monde, The Times of London, and the Dow Jones, Reuters, and AP newswires. For more information, see** http://www.factiva.com**.**
- **Hoover's, Inc, aggregates business information on 64,000 companies, including public, private, and non-US businesses. Hoover's capsules and profiles consist of company overviews, products, operations, officers, competitors, financials, and more. For more information, see** http://www.hoovers.com**.**
- **YellowBrix, Inc. is a leading provider of syndication services for enterprises and other Internet-connected business seeking to leverage**

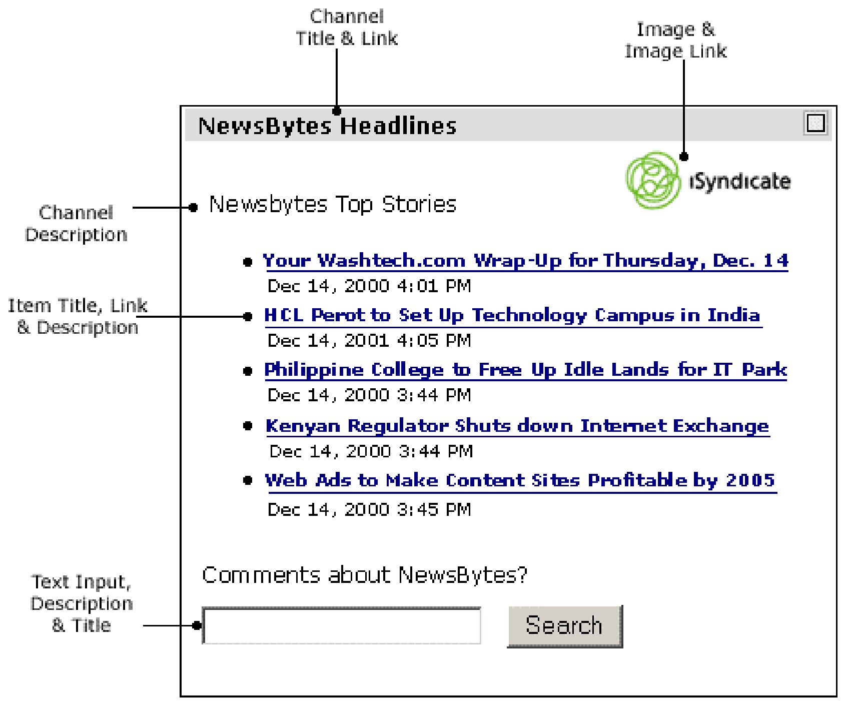
the Web for information collection and delivery. YellowBrix offers one of the largest selections of syndicated content available, including wireless content, from over 1200 leading brands and niche providers around the globe. For more information, see http://www.yellowbrix.com.

- ScreamingMedia, Inc. is a leading global provider of content solutions: content infrastructure, syndication, and services. ScreamingMedia aggregates licensed content, such as news, features, photos, video, stock quotes, audio, and weather reports, and then filters, delivers, and precisely integrates it into its customers' Web sites instantaneously. For more information, see http://www.screamingmedia.com.

Content syndication standards

Publication and subscription of content is supported in WebSphere Portal Server through OCS. The Open Content Directory Format provides a concise, machine-readable listing of syndicated channels. The directory format is capable of supporting multiple sites, each with multiple channels. Each channel can have multiple formats such as RSS versions 0.90 or 0.91, plain text, WML or Scripting News format as well as separate publishing schedules or languages.

WebSphere Portal Server also supports RSS to either broadcast changes in the portal pages or receive changes made in other channels. RSS is a lightweight XML format designed for sharing headlines and other Web content. Many content providers have been adopting RSS as a simple means of distributing headlines and links to new stories on their sites. RSS is becoming a vital *what's new* mechanism that helps attract users on the Web to the provider Web site. Some examples of the thousands of Web sites supporting RSS are: CBS, ZDNet, BBC, CNET, Rolling Stone, Forbes, USA Today, CNN, Disney, and AltaVista.



Newsbytes Top Stories

- Your Washtech.com Wrap-Up for Thursday, Dec. 14
  Dec 14, 2000 4:01 PM
- HCL Perot to Set Up Technology Campus in India
  Dec 14, 2001 4:05 PM
- Philippine College to Free Up Idle Lands for IT Park
  Dec 14, 2000 3:44 PM
- Kenyan Regulator Shuts down Internet Exchange
  Dec 14, 2000 3:44 PM
- Web Ads to Make Content Sites Profitable by 2005
  Dec 14, 2000 3:45 PM

Comments about NewsBytes?

Figure 9. RSS format

Search

WebSphere Portal Server offers a search service that supports distributed, heterogeneous searches across many data sources. It searches across all the data sources in parallel and combines the results into a unified list of matching documents. This federated search service is called Extended Search.

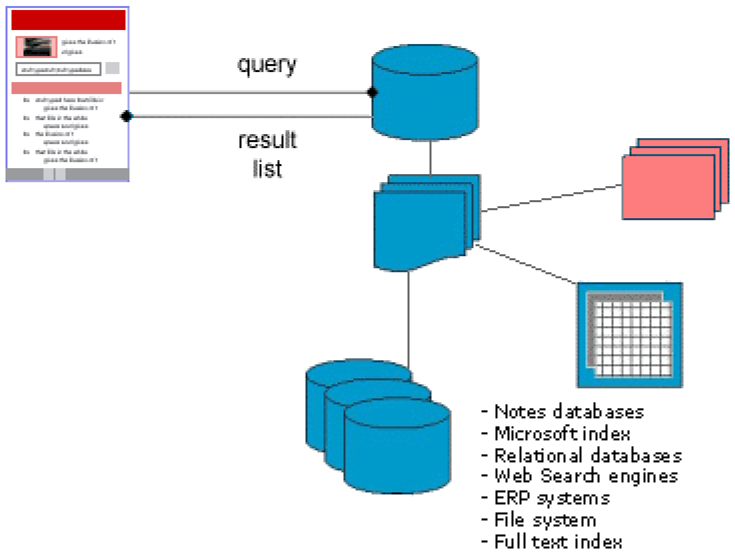


Figure 10. Search and indexing

With Extended Search, there is no need to create or maintain a central index. Extended Search translates each query into the native search syntax of the target data sources, thus hiding the complexity of the various query languages from the end user. All of the data sources are searched in parallel.

Searches can query and retrieve documents from repositories that include Lotus Notes 4.*x* and 5.x, and Domino.Doc. Extended Search also supports file systems, popular Web search engines, Microsoft Index Server, Microsoft Site Server, LDAP Directories, IBM Enterprise Information Portal databases, and relational databases such as IBM DB2®, Oracle, and other ODBC-compliant databases.

In summary, with Extended Search users can:

- **Search in parallel across Notes domains, legacy databases, local file systems, and popular Web search sites**
- **Get aggregated results presented as a single, ranked list of hits**
- **Save, reuse, and share searches**
- **Search across content in various languages and develop your applications in any language**
- **Refine search results to exclude documents that do not meet certain conditions (for example, exclude documents that were not created**

before a specific date or exclude documents that were not created by a specific author).

Site analysis

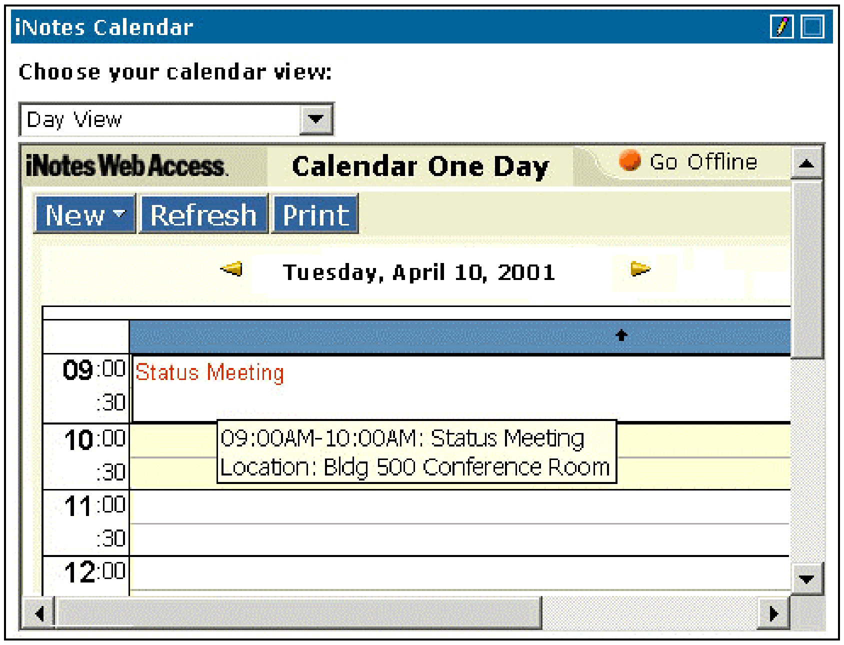
Site analysis provides information on Web site visitor trends, usage, and content. WebSphere Site Analyzer detects visitor trends and preferences, manages Web site content and structure, and improves the overall effectiveness of Web initiatives and campaigns. WebSphere Site Analyzer is the only Web analytic software available that is designed to be tightly integrate across the WebSphere software platform. WebSphere Site Analyzer can be used to track portlet usage and performance. Usage analysis can then be categorized to gather statistics about specific groups of information. WebSphere Site Analyzer is included with select WebSphere Portal Server solutions.

Collaboration

Collaboration starts as e-mail access and grows into team rooms, instant messaging, and communities of interest. Collaboration is provided in WebSphere Portal Server through a rich set of portlets and through integration with Lotus® K-station™, a knowledge portal with collaborative functionality.

Portlet support

WebSphere Portal Server can use e-mail, calendar, and scheduling portlets for Notes, iNotes™, and Exchange. Both iNotes and standard Notes servers are supported. Exchange portlets give access to Microsoft Exchange e-mail, calendar, address book, journal, and to-do list. The following figure shows an example of an iNotes Calendar portlet as displayed on a portal page.



Status Meeting

09:00AM-10:00AM: Status Meeting
Location: Bldg 500 Conference Room

Figure 11. iNotes portlet

Portlets for Sametime and QuickPlace provide instant collaboration. The Sametime portlet offers real-time collaboration through instant awareness, communication, and screen sharing capabilities. In addition, Sametime includes a comprehensive application development toolkit that enables you to embed real-time capabilities, like live expert links or real-time help features, into your existing applications. The QuickPlace portlet provides a team workspace in the portal. Teams use QuickPlace to share and organize ideas, content, and tasks around any project or ad-hoc initiative. QuickPlace provides a central on-line workspace structured for productivity.

K-station integration

Lotus K-station can integrate with WebSphere Portal Server so that the collaborative capabilities provided with the K-station interface work together with the framework provided by WebSphere Portal Server. The unique capabilities built into K-station allow users to see the following things at a glance:

- **Who is on-line and their availability for instant conversations**
- **Who is currently working in the same Community Place**
- **A list of members that belong to a specific Community Place**
- **A personal buddy list of other users**

Also, any portlets created with the WebSphere portlet API can display within the K-station user interface.

Enterprise application integration

WebSphere Portal Server supports the integration of new applications or legacy applications. You can use portlets that support WebSphere development and integration services, and you can use Application Access Feature which provides connectors between the portal and enterprise applications.

Integration development

Portlets written in Java code or JSP can integrate with elements of the WebSphere Application Server programming model. This programming model includes comprehensive application development and integration services:

- **J2EE is the glue between the application server and various enterprise applications. It offers broad functionality in the areas of application packaging, object services, transaction services, programming standards, and Java messaging services.**
- **XML provides a simple and nearly universal data representation. WebSphere has strong XML support, including XML parsing, XML style sheet transformation, and SAX-event based XML parsing.**
- **Middleware connectors provide access to many enterprise application systems through Java bean interfaces. MQSeries® Application Integrator and WebSphere Java Connector Extension (J2C) provides a full suite of application integration connectors, including connectors for**

CICS ECI/EPI, Encina DE-Light, IMS ITOC, MQSeries, Host-on-Demand, and SAP R/3.

Web applications that already use these middleware services can be reused directly in the portal via the URL portlet. In other words, portlets are able to display existing Web page markup such as HTML or XML. In this way, WebSphere Portal Server enables reuse of existing Web assets.

Application Access Feature

Included with WebSphere portal solutions is the WebSphere Portal Server Application Access Feature. The WebSphere Portal Server Application Access Feature enables WebSphere Portal Server to access structured data in enterprise applications. By installing the optional Application Access Feature, you gain the ability to add data from your enterprise applications (such as SAP, Baan and Siebel) to portal pages. Connectivity to the enterprise applications is enabled by specialized connectors. Data retrieval is enabled by custom applications (such as, SAP Portals iViews for Baan) that are converted to portlets and then added to portal pages. You can select from the extensive list of predefined custom applications or create your own.

Web services

A Web service is an interface that describes a collection of network accessible operations. A Web service is described using an XML description language, so that the service can be invoked without prior knowledge of the platform, language, or implementation design of the Web service.

In a future release, WebSphere Portal Server will provide support for Web services. Portlets will be able to use Web services to perform their processing, and portal administrators will be able to bind remote portlets as Web services, making the remote portlets available in the portal registry dynamically.

For example, a large corporation might have several different portals, such as an employee portal, a supplier portal and a human resources portal. Each of these portals may choose to publish some of its portlets as Web services for access through other the portals.

Individual portlets can also bind to Web services in delivering their functionality. For example, a search portlet might query the user for a search string, then use a search Web service to search the Internet. Or, calendar portlet might act as a front end, providing views for a calendar Web service.

WebSphere Portal Family solutions

WebSphere Portal Server ships within the WebSphere Portal Family solutions. Because the software components are different for each solution, the portal services that are available vary.

The entry portal solution, called WebSphere Portal Enable, enables utilization of WebSphere Portal Server, WebSphere Personalization, and WebSphere Application Server. WebSphere Portal Enable provides a number of portlets for e-mail, calendars, collaboration, syndicated news, and other functions. In keeping with IBM's open product strategy, a portlet API is available so customers or other software vendors can extend the framework.

The mid-level portal solution, called WebSphere Portal Extend, builds upon the framework and services provided by WebSphere Portal Enable. By providing K-station integration, Sametime, and QuickPlace, WebSphere Portal Extend enables a fully-functional collaborative portal. In addition, Domino™ Extended Search and Site Analyzer are included to provide search services and site analysis, respectively.

The top-level portal solution, called WebSphere Portal Experience, builds upon the services provided by WebSphere Portal Extend. Additional security services are provided through Policy Director, and content-management services are provided through IBM Content Manager and IBM EIP Client Kit.

Portal Infrastructure

This section provides information about the internal features included with WebSphere Portal Server, such as user and group management, and security. Also, information about how the portal interacts with new and existing systems within an enterprise is discussed.

User and group management

WebSphere Portal Server provides Web pages that allow users to enroll at the portal and to self-manage their own preferences and account information. Alternatively, enterprises can integrate the portal with existing user directories, and may choose to disable the self-enrollment pages.

Federated user profile

WebSphere Portal Server provides connectivity between the portal and information in various user directories.

Directory standards

By default, the LDAP directory stores user-specific data as entries that are associated with the *inetOrgPerson* schema.

- User-specific data; such as the user name, user ID, and password is stored in an LDAP directory. The Java Naming and Directory Interface (JNDI) enables read/write interoperability between WebSphere Portal Server and the LDAP directory.
- Portal-specific data, such as home page settings and portlet settings, is stored in a relational database management system (RDBMS). WebSphere Portal Server supports IBM DB2 and Oracle 8*i*.

WebSphere Portal Server provides a Java bean interface for accessing user information. The User bean acts as an interface to a stateless session Enterprise JavaBean (EJB), which in turn, acts as a consolidation interface to multiple back-end EJB classes, each responsible for retrieving a portion of the user data. The following figure shows the general structure.

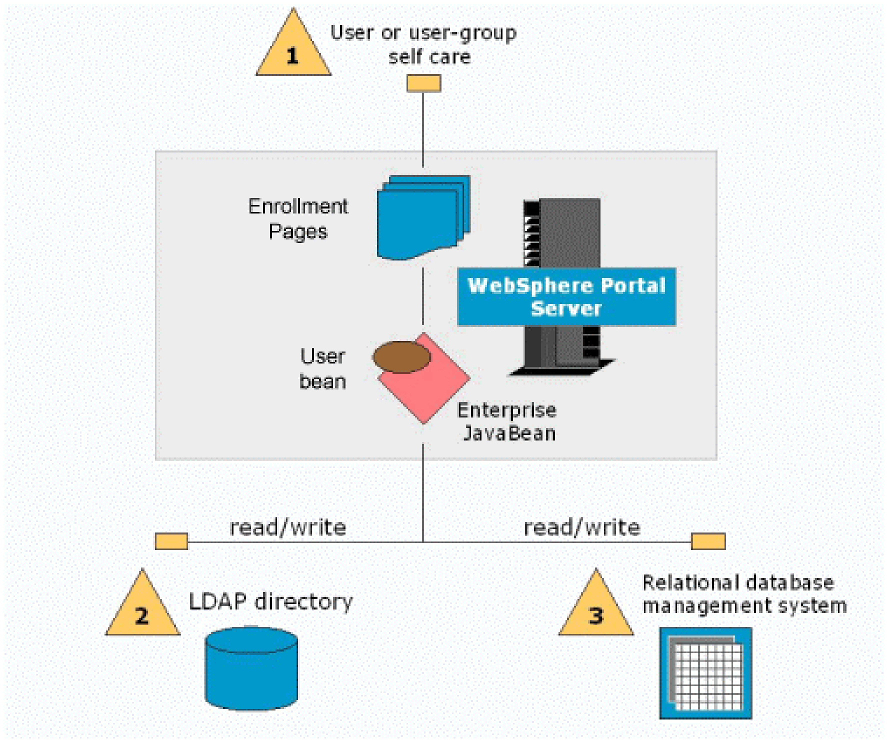


Figure 12. Federated user profile

User, group, and access beans

The portal server provides a Java bean interface for users, group memberships, and access control.

The back-end implementations can be replaced to support existing infrastructure or third-party products.

1. The user and group self care interface begins with the portal enrollment pages. These pages can be modified to collect additional attributes or to allow the user to specify group membership information.
2. The User bean class calls the backing EJB to store or retrieve the basic user information (such as name or city) in the LDAP directory.
3. The bean also calls another EJB to store or retrieve the portal-specific settings for the user, such as the user's list of portlets and their settings.

The EJB implementation classes can be exchanged transparently, so that third-party LDAP servers, alternate schemas, and other external data stores can be easily integrated into the portal user-management system. Each customer can replace the implementation classes to match where their data is stored. The source code for the default implementation of WebSphere Portal Server is provided as an example.

Existing directories

Because many companies already have databases that contain user data, WebSphere Portal Server enables the federation of user data from multiple, existing repositories. For example, in many business-to-employee portals, a directory of user information already exists, so it is desirable to access user data in a read-only LDAP directory and then store additional user data in a relational database tables.

In this scenario, illustrated in the following figure, user self-registration is disabled, so WebSphere Portal Server does not update the LDAP directory at all.

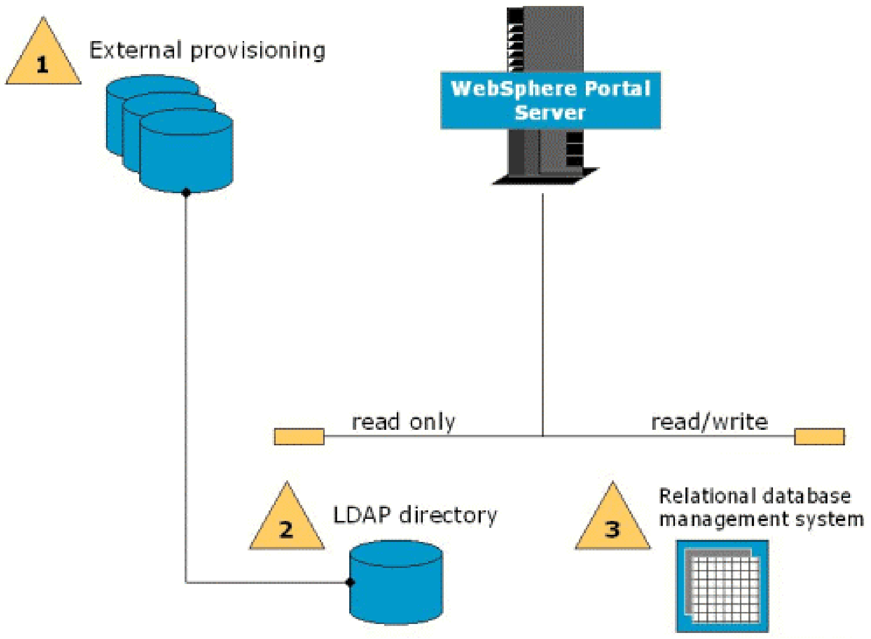


Figure 13. External provisioning

Another important scenario involves the use of WebSphere Everyplace Server, which includes Tivoli® Personalized Services Manager (TPSM), an integrated subscriber management system. When TPSM is used, the architecture is usually a hybrid of the two scenarios outlined so far.

The hybrid scenario uses self-enrollment pages to populate the TPSM repository. TPSM provisions the LDAP directory, and the portal server does not update LDAP at all. The implementation details of this scenario are isolated in the implementation classes of the user EJB, but the User Java bean interface remains unchanged.

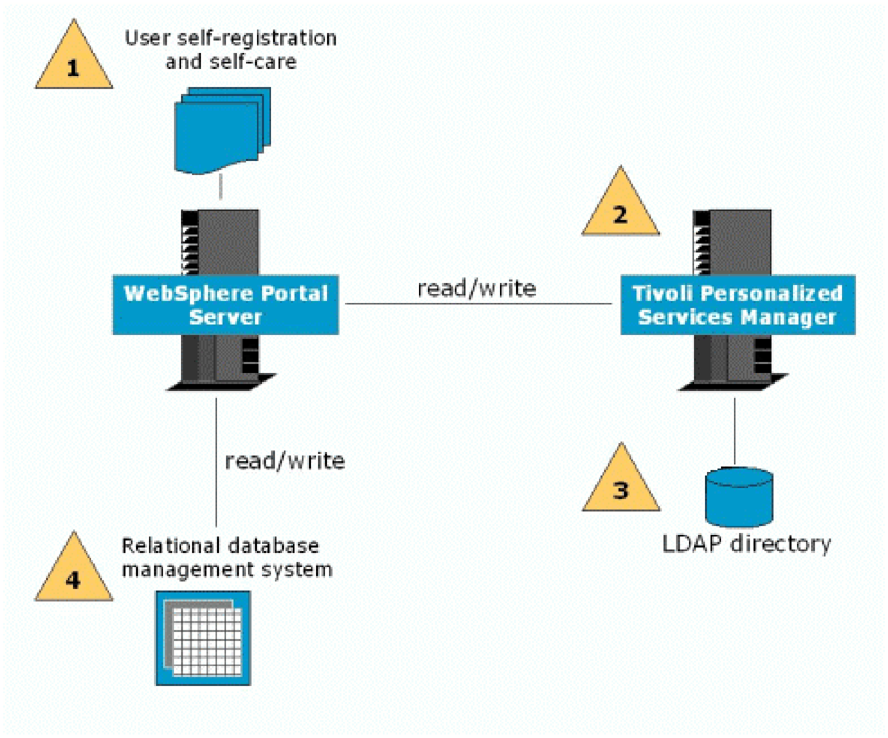

1
User self-registration and self-care
2
WebSphere Portal Server
read/write
Tivoli Personalized Services Manager
read/write
3
LDAP directory
4
Relational database management system

Figure 14. Advanced subscriber management

Group management

The user bean class also gives access to group information. Users may be classified into one or more groups. Group membership information is stored in an LDAP directory, and the group names are defined using the LDAP directory administration interface.

Delegated administration of users is through the LDAP administration interface. Access control permissions are assigned through a policy specification portlet interface (or through the administration interface of a third party authorization server, if applicable). The portal server protects access to portlets, and the backing LDAP server protects access to users and groups.

Security

WebSphere Portal Server uses standard Java Security APIs for its authentication, authorization, and single sign-on features. The default implementations can easily be replaced to support existing infrastructure or third-party products.

Authentication choices

WebSphere Application Server security is included with all WebSphere portal solutions.

The WebSEAL-Lite authentication component is shipped with select WebSphere portal solutions.

Authentication

The portal server should be configured so that incoming requests pass through an authentication component such as WebSphere Application Server security; WebSEAL-Lite, which is a component of Policy Director; Netegrity SiteMinder; or other authentication proxy servers. A separate alias is configured to allow anonymous requests for new users or for users who have not yet logged in.

Authentication proxy servers can be integrated with WebSphere Application Server through its Trust Association Interceptor APIs. This provides a highly secure and uniform interface to WebSphere Portal Server. Examples of authentication proxy servers that can be supported this way include IBM WebSEAL-Lite and WebSphere Everyplace Server Authentication Server.

WebSphere security and the IBM authentication proxy servers are configured to use the portal LDAP directory to authenticate users. Once the authenticated user information is available, WebSphere Portal Server stores various credentials, including LTPA tokens, CORBA credentials, user id and password, etc. These credentials are available to portlets through a standard JAAS API interface, so that they can be passed to back-end applications to achieve single sign-on. This avoids having the user prompted again for authentication.

Another technique for integrating third-party authentication servers is to replace the portal server login action class. The replacement login action would inspect the HTTP header information (depending on the details of the third-party product) to handle the login.

Authorization

The access control interface of the portal integrates with the user and group beans to find out which portlets a user is authorized to use. The aggregation modules and the page customizer use this information to filter the list of portlets that are displayed.

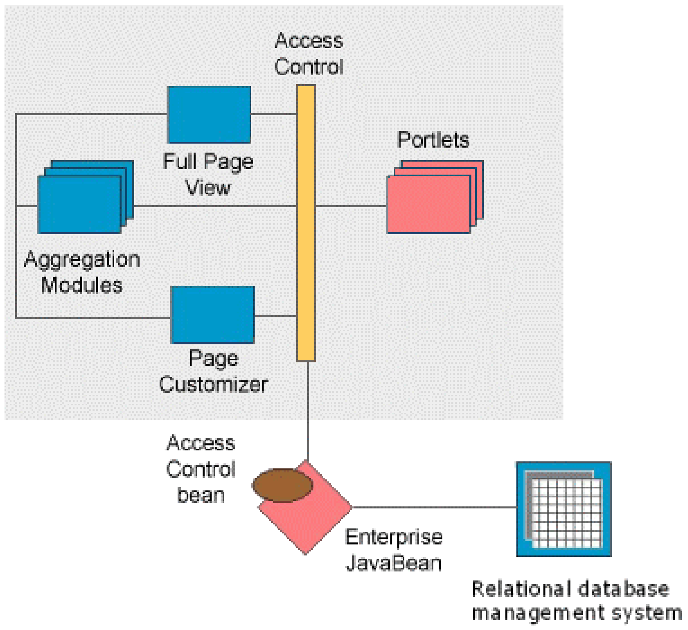


Figure 15. Access control

The access control system consults the policy engine to determine access rights; access control lists are stored in database tables. Administrators can define access control list that manages access to each portlet. The administrator interface is itself a portlet whose access is managed through these same access control lists.

The access control model is additive, meaning that a user can be granted access to a resource either directly or indirectly (as a result of granting access to one of the user groups). The priority or hierarchy of group membership has no bearing on access control rights; if a user is granted access by any means, then access is granted. For example, if an access control list is changed so that a portlet is removed from a higher-level group, this would not revoke access rights for a specific user if access had been granted to the user through some other group membership.

By replacing the access control system implementation classes, third party access control servers (such as Policy Director or Netegrity SiteMinder) may be used instead of the default WebSphere Portal Server access control implementation. Source code is provided to make this replacement easier. In such cases, the third party access control system would provide its own administration interface for managing the access control lists.

Page transformation

As a separately purchased product, IBM offers the WebSphere Transcoding Publisher, which can transform the markup produced by WebSphere Portal Server to target additional devices. WebSphere Transcoding Publisher can be used in several different configurations:

- It can be installed in a proxy configuration, so that it transforms outbound markup before the markup is sent to the browser. This configuration is useful when making the portal accessible through personal digital assistant (PDA) devices, for example. It is also useful for WML/WAP devices, where the payload size is very limited, since WebSphere Transcoding Publisher can split a large WML deck into several smaller ones.
- Another useful way to use Transcoding Publisher is to develop a portlet that calls the transcoding service during page aggregation. This way, individual portlets can produce their mark-up automatically, rather than requiring custom JSP views or style sheets for each mark-up. This structure is particularly useful for rapidly changing content, such as news, or for clipping content from existing HTML pages. An advantage of this configuration is that only specific portlets perform the transformation step, rather than always processing the entire page.
- In a very similar manner, WebSphere Translation Server can be configured to transform content into other human languages, such as transforming English to Spanish, for example.

Performance

This section discusses performance considerations for the portal, such as typical configurations, load balancing, and content caching.

Typical configurations

The simplest configuration for prototyping or proof-of-concept installations of WebSphere Portal Server is quite minimal. An evaluation installation requires only WebSphere Application Server (Advanced Edition), IBM SecureWay® Directory, and DB2, all running on a single server. The WebSphere Portal Server installation program installs everything else that you need, including WebSphere Personalization.

Larger installations would include the same basic components load balanced across several production servers for greater reliability and scaling.

Additional infrastructure

WebSphere Everyplace Server provides additional infrastructure for service providers and enterprises. Together WebSphere Everyplace Server and WebSphere Portal Server deliver a portal to wireless devices such as phones and personal digital assistants operating over a variety of wireless networks such as GSM and GPRS. WebSphere Everyplace Server provides a secure and scalable framework for network connection, authentication, device and subscriber management, data transformation, load balancing, and caching. The following figure shows an example of a typical configuration.

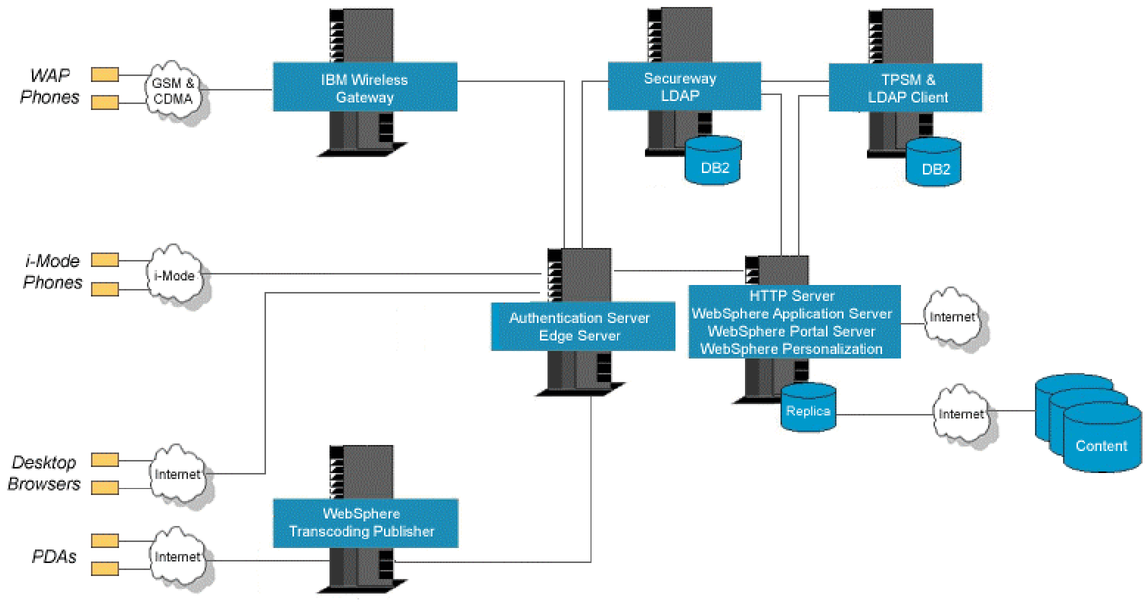


Figure 16. WebSphere Everyplace Server Configuration

Content caching

WebSphere Portal Server can work with live content from the Internet. For better performance, it uses a cache to keep local copies of remote documents. The disk cache contents are refreshed either at an interval set by the portal administrator, or based on an interval specified in the document HTTP header. OCS channels are also cached. The channel

entries specify the frequency and period for updating content for each channel.

Load balancing

WebSphere Portal Server runs in a WebSphere Application Server cluster to achieve scalability and reliability.

The facilities of WebSphere Edge Server can also be used for additional load balancing and high availability. To achieve optimal fail-over, persistent sessions are used, storing all session data in a shared database. If one portal server fails, the Edge Server network dispatcher component will detect the situation and will balance further requests between the remaining portal servers.

To achieve optimal performance, it is best to always route requests from a single client session to the same server, so that the session data is retrieved more efficiently. In practice however, proxy servers interfere with the load balancing because they collapse IP addresses, and replace the true IP address of the client with the address of the proxy server. The result is that all clients connecting via the same proxy would be routed to the same portal server all the time.

To achieve *sticky* **sessions with reasonable load balancing, the network dispatcher component can be set up in a special way. It is configured for load balancing without sticky sessions on initial requests. When the first request from a client is received, the server redirects the client back to itself, thus bypassing the network dispatcher. The client directly communicates with the server and the session context need not be shared, as a particular client will always talk to the same server.**

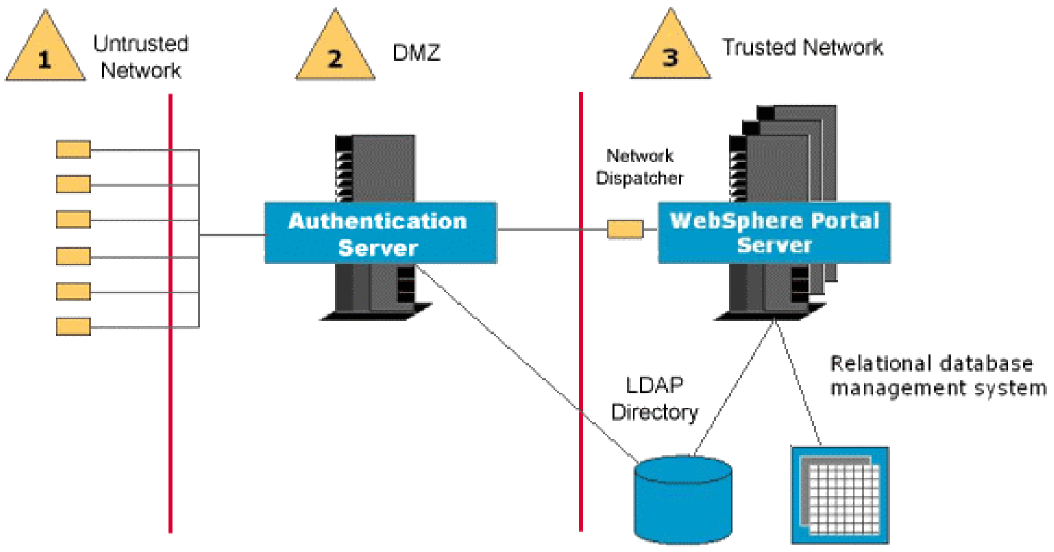


Figure 17. Load balancing

Of course the direct communication between client and server will not work with firewalls and an authentication proxy in the demilitarized zone. So an alternative configuration has initial requests from clients being load balanced by the network dispatcher. Portal servers then force sticky sessions by redirecting the client to their own alias.

Summary

The WebSphere Portal Server framework simplifies many of the tasks involved in building complex portal Web sites. It targets multiple devices and helps companies leverage and reuse existing their Web assets.

WebSphere Portal Server provides a scalable, secure, and extensible infrastructure for building a wide range of portals, including business-to-consumer, business-to-employee, and business-to-business portals. IBM offers many other products that complement WebSphere Portal Server:

- WebSphere Personalization provides users with the capabilities to build a portal that delivers pages that are customized to the interests and needs of each site visitor.
- Lotus K-station provides the tools to access, organize, and share information, and to collaborate with people and teams.
- Lotus Sametime provides instant messaging, shared white boards, and application sharing for electronic meetings, which are accessible through portlets. Lotus QuickPlace provides team workspaces for sharing and organizing ideas, content and tasks.
- Lotus iNotes™ Web Access provides a Web interface to Lotus Notes e-mail, calendar, address book and to-do lists. WebSphere Portal Server provides portlets for each of these functions.
- Domino Extended Search provides a high performance search capability across Internet document sources.
- IBM Content Manager and IBM EIP Client Kit can index, store, search, and distribute digital content, including business documents, printed reports or statements, and audio/visual, text, XML, and HTML files.
- SecureWay Policy Director provides security infrastructure, including authorization and authentication services.
- WebSphere SiteAnalyzer is useful for measuring the activity and effectiveness of the portal. It reports on structural information, such as broken links and page sizes. It also processes Web server logs to reveal how the portal site is being used, who is using it, where they enter or exit and how they navigate within the site. The information is mined and stored in a database or displayed in reports.
- WebSphere Studio and VisualAge for Java are productive development tools for working with the portal. WebSphere Studio is based on Eclipse, the Java-based open source software.
- WebSphere Everyplace Server includes infrastructure and services for large portal installations. It provides a secure and scalable framework for network connection, authentication, device and subscriber management, data transformation, load balancing, and caching. Included is WebSphere Edge Server which provides load balancing and content caching services needed for highly scalable and fault-tolerant portal installations. Also included is WebSphere Transcoding Publisher which adapts and optimizes content for new mobile devices or other browser environments. In particular, it is a useful addition for customers who are using WebSphere Portal Server mobile device support because it optimizes the deck structure of WML portals.