

Portal solutions
May 2005



WebSphere software

A guide to IBM WebSphere Portal, Version 5.1.

Contents

2	<i>Introduction: The portal vision</i>
3	<i>IBM WebSphere Portal: A leading-edge portal platform</i>
5	<i>WebSphere Portal architecture</i>
8	<i>Portlets</i>
10	<i>Portlet applications</i>
10	<i>Portlet modes</i>
11	<i>The portlet API</i>
11	<i>Portlet performance</i>
12	<i>Standards</i>
13	<i>Portlet cooperation</i>
14	<i>Brokered cooperation</i>
15	<i>Discoverable services</i>
16	<i>Development tools</i>
18	<i>Content management</i>
20	<i>Document management</i>
22	<i>Web content management</i>
25	<i>Productivity components</i>
30	<i>Security</i>
41	<i>Personalization</i>
44	<i>Cascading authorization for portal customizations</i>
45	<i>Skins and themes</i>
49	<i>WebSphere Portal release builder</i>
49	<i>Virtual portals</i>
50	<i>Universal access</i>
52	<i>Using portlets to administer your portal</i>
64	<i>Collaborative services</i>
66	<i>Business process integration and task management</i>
68	<i>Dynamic UI configurations</i>
69	<i>IBM WebSphere Commerce</i>
69	<i>Mobile portals</i>
71	<i>Supporting new devices</i>
74	<i>A complete platform for collaboration and enterprise application integration</i>
74	<i>For more information</i>

Introduction: The portal vision

As a core component of IBM® Workplace™ technology, the front end of computing that strives to make people more productive in the context of the business they do every day, IBM WebSphere® Portal for Multiplatforms, Version 5.1 helps simplify your organization's technology infrastructure by offering a single point of personalized interaction with applications, content, processes and people. To deliver a unified user experience, and an innovative, adaptive user environment designed to transform productivity, WebSphere Portal brings together a range of leading-edge technologies designed to give you a flexible, open, extensible framework to build successful business-to-employee (B2E), business-to-business (B2B) and business-to-consumer (B2C) portals.

This leading-edge family of products, technologies and solutions enables you to customize work environments based on users' unique roles in your organization. WebSphere Portal software can help reduce costs by leveraging existing investments in enterprise applications, data and security technology – while you evolve to meet the challenges of today's on demand business environment.

WebSphere Portal software is an open-standards-based framework supporting a range of options across databases, directories, platforms and security, with portlets serving as a key component. The term *portlet* refers to a small portal application, usually depicted as a small box in the Web page. Portlets are reusable components that provide access to enterprise applications, Web-based content, host and data systems, content-management systems, process-driven workflow applications, and other resources. Web pages, Web services, applications and syndicated content feeds can be accessed through portlets. WebSphere Portal software includes pre-integrated portlets, cross-portlet integration for all application types, tools that enable you to create new portlets easily and the ability to construct composite applications to manage business processes and workflow transactions spanning multiple enterprise systems. As a result, WebSphere Portal helps organizations move beyond fragmented application silos while hiding the complexity of the IT infrastructure. These advantages help improve employee productivity and business responsiveness, help cut costs and strengthen relationships with customers and partners.

This white paper is intended to help IBM clients, independent software vendors (ISVs) and application architects plan their use of WebSphere Portal. It explains a range of WebSphere Portal features, including portal application and integration capabilities, security, user management, administration, document management, content publishing, search, personalization, collaboration, and process integration. IBM envisions portals as the key to providing a personalized, relevant Web experience, enabling users to readily find the people, processes and information relevant to their roles – and the ability to manage the selection and personalization of content to suit their changing needs.

IBM WebSphere Portal: A leading-edge portal platform

WebSphere Portal uses the advantages provided by the IBM WebSphere software platform. The WebSphere software platform delivers standards-based integration and infrastructure software to maximize business flexibility and responsiveness. As Figure 1 shows, the WebSphere software platform is built with services to support scalable, reliable Web applications, with components and technologies that enable you to add extensions to new applications and processes, and provide integrated collaborative services.

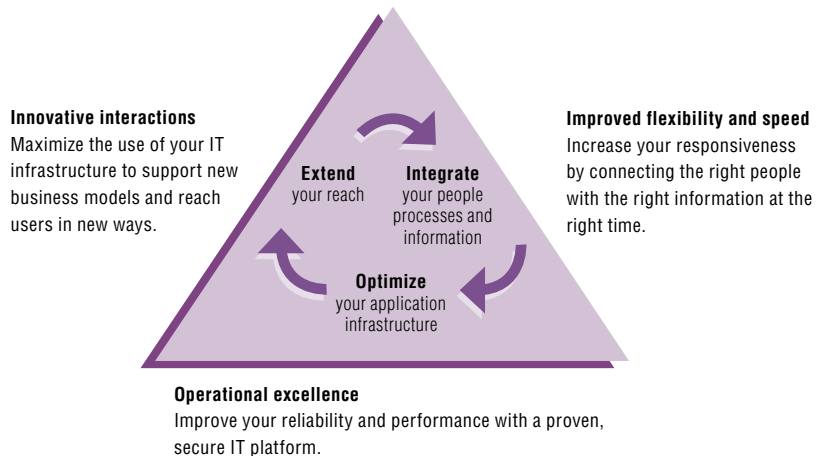


Figure 1. WebSphere software for on demand business

The WebSphere software platform capabilities can be grouped into five areas of functionality.

People integration

Interact with information, applications and business processes at any time from anywhere to personalize Web-based content and to make it accessible to any device. WebSphere products, such as WebSphere Portal, fine-tune your users' experiences and provide broad access for your customers, employees, business and trading partners, and remote branch offices.

Process integration

Optimize and integrate internal business processes – or even processes that involve your business partners – to keep them in line with strategic goals. WebSphere offerings such as WebSphere Business Integration Server Foundation make it easy for your company to implement applications and business processes that can streamline supply chain management (SCM) and help you enable existing processes for the Web.

Information integration

Access and manage information that is typically scattered throughout the enterprise and across the value chain. WebSphere offerings such as IBM WebSphere Information Integrator manage storage, organization and access to structured and unstructured content and data sources across geographical locations.

Application integration

Helps ensure reliable and flexible information flow between diverse applications and organizations. WebSphere software supports standards-based communication protocols and application adapter interfaces, enabling reliable and seamless exchange of data between multiple applications.

Application infrastructure

Build, deploy, integrate and enhance new and existing applications. Help ensure a reliable enterprise processing infrastructure, extend legacy applications and logic and modernize interfaces for use in new Web environments.

WebSphere Portal software provides an extensible framework for interacting with enterprise applications, content, people and business processes. It includes IBM WebSphere Application Server and WebSphere Business Integration Server Foundation components to support scalable Web application server and business-process integration solutions managed from the portal framework. WebSphere Portal self-service features enable users to personalize and organize their own view of the portal, to manage their own profile, and to publish and share documents with their colleagues.

WebSphere Portal architecture

By integrating key IBM products and technologies, and providing application programming interfaces (APIs) and extension points for Business Partners and clients to extend and customize their environments, WebSphere Portal has become the industry's most comprehensive portal solution, and represents the de facto standard on demand business architecture (see Figure 2).

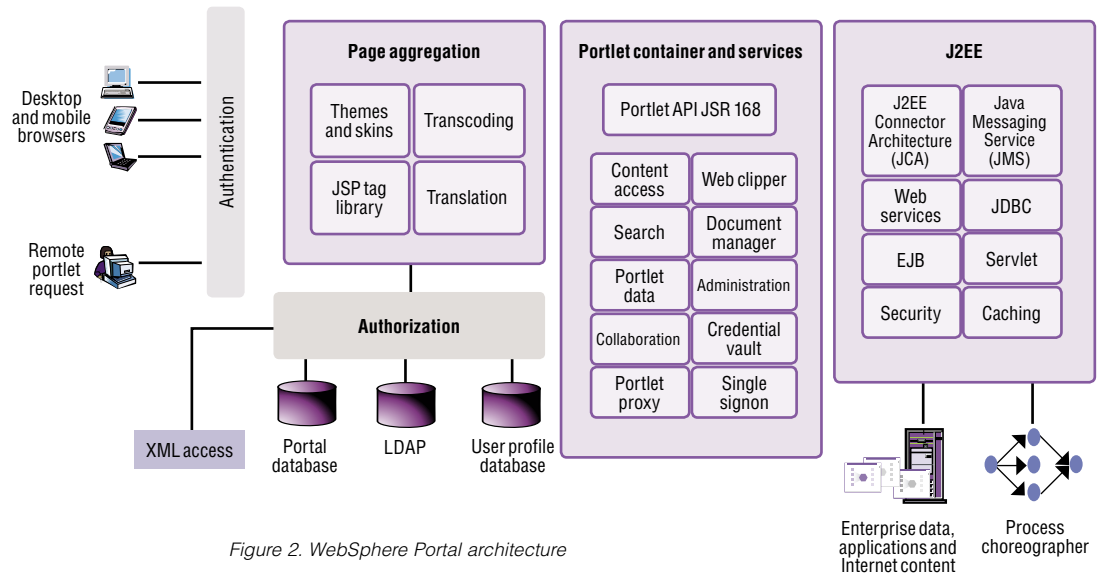


Figure 2. WebSphere Portal architecture

The more fully you integrate and leverage your work environment, the more you can embrace on demand business principles. An *on demand business* is an enterprise where business processes, integrated end to end across the company and with key partners, suppliers and customers, can respond with speed to almost any customer demand, market opportunity or external threat. More than operational efficiency, on demand business is about building a dynamic infrastructure that allows you to integrate, modify and leverage existing applications and processes cost-effectively.

As Web-based applications enter the era of on demand business, WebSphere Portal leads the transformation with a variety of capabilities, including delegated administration, cascading page layouts and portal federation through Web services. It also includes support for standards, advanced portlet application concepts, process integration, task management and search services. Complementary offerings enable enhanced mobile functions, such as intelligent notification, offline browsing and data synchronization.

Other offerings that use the services of the WebSphere Portal foundation platform are also available. These include IBM® Workplace™ Services Express, a team-collaboration and document-management environment for small-to-midsized deployments. Another offering is IBM® Workplace™ Collaboration Services, a solution designed to provide a platform for advanced team collaboration, messaging and chat services, and activity-explorer, document-management and learning applications. IBM® Workplace™ Collaboration Services provides a choice of client experiences, ranging from standard browsers to rich-client applications, and uses a network-oriented application deployment model, improving administrative and user implementation processes. IBM® Workplace™ offerings also include industry-specific solutions that can offer preconfigured capabilities built to support user and team productivity, and business-management capabilities, which take advantage of supporting WebSphere Portal application-integration framework services.

IBM WebSphere Portal for Multiplatforms, Version 5.1 is available in several editions (see Table 1), each designed to provide the infrastructure you need to build and deploy highly scalable portals. All offerings share a common framework (the portal server), and might share certain products and services. The portal server provides common services, such as application connectivity, integration, administration and presentation that are required across portal environments. To get the latest details about supported platforms, system requirements and version numbers, visit ibm.com/software/genservers/portal/.

IBM WebSphere Portal Enable

Model	Advantage
IBM WebSphere Portal server	Provides presentation, user management, security and other services for constructing a portal.
IBM® WebSphere Portal personalization engine	Includes advanced personalization technologies to target Web content to meet user needs and preferences.
IBM® Workplace Web Content Management™	Provides tools for authoring, personalizing and publishing content and documents contributed to the portal.
IBM WebSphere Portal Document Manager	Centralizes document storage, organization and version-management services.
Productivity components	Offers an inline view and edit capabilities designed to support rich text, spreadsheet and presentation content.
IBM Tivoli® Directory Server	Provides a Lightweight Directory Access Protocol (LDAP) server.
IBM WebSphere Translation Server	Delivers translation services that enable Web content, e-mail and chat content conversions across languages.
IBM Rational® Application Developer	Provides professional developer tools to create, test, debug and deploy portlets, servlets and other assets related to portals and Web applications.
IBM Lotus Collaboration Center	Includes preconfigured pages and portlets to deliver out-of-the-box portlet access to IBM Lotus® Domino® and extended products, including IBM Lotus Notes® and Domino mail and applications, IBM Lotus Domino Document Manager, IBM Lotus Instant Messaging (Sametime®), IBM Lotus Team Workplace (QuickPlace®), and IBM Lotus Web Conferencing Server.
Collaborative components	Includes Java™ application programming interfaces (APIs) that provide the building blocks for integrating the functionality of Domino, Lotus Instant Messaging and Lotus Team Workplace into portals and portlets.

WebSphere Portal Extend

All products in WebSphere Portal Enable	Includes the same portal server and personalization functions as WebSphere Portal Enable.
IBM Lotus Extended Search	Delivers parallel, distributed, heterogeneous searching capabilities across Lotus Notes and Domino databases, Microsoft® sources, relational database management system (RDBMS) data stores, Web search sites and other sources.
IBM Tivoli Web Site Analyzer	Analyzes Web-site usage logs to reveal information that you can use to improve your portal to provide better user experiences.
Lotus Instant Messaging	Offers instant messaging, presence awareness and Web-conferencing services.
Lotus Team Workplace	Provides a Web-based solution to create team workspaces for collaboration. Features include discussions, document collaborations, and the ability to coordinate plans, tasks and resources.

Table 1. WebSphere Portal for Multiplatforms, Version 5.1 product editions

Choosing a WebSphere Portal offering can help you realize tangible business and technical benefits, including:

- *Improved revenue-generating capabilities as a result of tighter relationships with customers or partners, workforce productivity, innovation and reduced cycle times*
- *Reduced operational costs as a result of improved operational efficiency, better information flow and knowledge, and a consistent infrastructure*
- *Increased employee productivity and improved decision making as a result of providing access to more-relevant information and offering a single access point to collaborative services, applications and processes*
- *A better user experience as a result of improved authentication, security and single signon capability*
- *Reduced training costs as a result of a common presentation and a consistent user interface*
- *An extended, more useful life for applications and processes as a result of unifying their presentation and providing a means of accessing them through desktop and mobile devices*

The WebSphere Portal offerings allow an on demand business to quickly capitalize on its IT and human assets while presenting a first-class Web experience to its employees, partners and customers.

Portlets

You can select from portlets provided in the WebSphere Portal release, including prebuilt portlets and Web and application integration portlet tools. You can also choose from a catalog of portlets created by IBM and by IBM Business Partners, or you can create your own portlets.

Any particular portlet is developed, deployed, managed and displayed independent of other portlets. Administrators and end users create personalized portal pages by choosing and arranging portlets, resulting in Web pages that present content tailored for individuals, teams, divisions and organizations (see Figure 3).

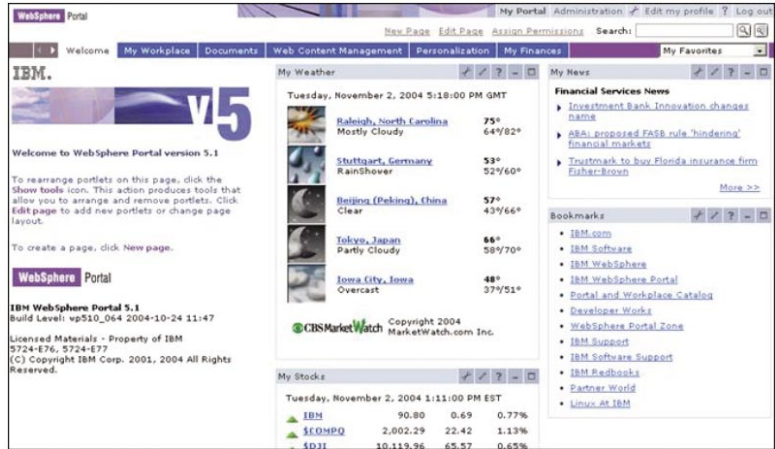


Figure 3. A typical portal page

The portal server includes a rich set of standard portlets for storing and sharing documents, displaying syndicated content and performing XML transformation. It also includes portlets for accessing existing Web pages and data applications, Lotus Notes and Microsoft Exchange productivity applications, Lotus Instant Messaging and Lotus Team Workplace applications. IBM also offers new and updated portlets and solutions from IBM and IBM Business Partners, called the *IBM WebSphere Portal Catalog*. In Figure 4, you can see just a few of the portlets available from IBM and IBM Business Partners through the catalog. You can visit the catalog Web site at ibm.com/websphere/portal/catalog.

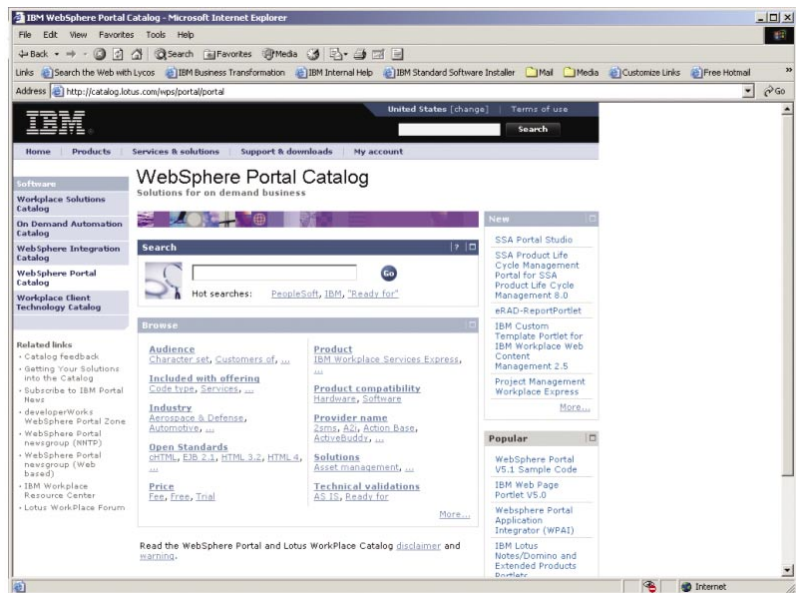


Figure 4. IBM WebSphere Portal Catalog

Portlet applications

Portlets are more than simple views of existing Web content. A portlet is a complete application, following a standard model-view-controller (MVC) design. Portlets have multiple states and view modes, as well as event and messaging capabilities. Portlets run inside the portlet container of a portal server, similar to a servlet running on an application server. The portlet container provides a run-time environment in which portlets are instantiated, used and, finally, destroyed. Portlets rely on the portal infrastructure to access user-profile information, participate in window and action events, communicate with other portlets, access remote content, search for credentials and store persistent data.

Generally, you can administer portlets more dynamically than servlets. For example, you can install or remove portlet applications consisting of several portlets while a server is running. You can also change the settings and access rights of a portlet while the portal is running, even in a production environment.

Portlet modes

Portlet modes allow a portlet to display a different user interface, depending on the task required of the portlet. A portlet has several modes of display, which can be invoked by icons on the portlet title bar. These modes include *view*, *help*, *edit* and *configure*.

A portlet is initially displayed in its *view* mode. As the user interacts with the portlet, it might display a sequence of view states, such as forms and responses, error messages, and other application-specific states. The *help* mode provides user assistance about the portlet. *Edit* mode provides a page for users to change the portlet settings. For example, a weather portlet might include an edit page so that users can specify their location. Users must log into the portal to access edit mode. If *configure* mode is supported by a portlet, it provides a page for portal administrators to configure portlet settings that are shared by instances of that portlet.

Each portlet mode can be displayed in *normal*, *maximized* or *minimized* states. When a portlet is maximized, it is displayed in the entire body of the portal page, replacing the view of other portlets. When a portlet is minimized, only the portlet title bar is displayed on the portal page.

The portlet API

Portlets are a special subclass of the `HttpServlet` class, with properties that allow them to easily plug into and run in the portal server. They are assembled into a larger portal page, with multiple occurrences of the same portlet displaying different data for each user. The portlet API provides standard interfaces for portlet functions. It defines a common base class and interfaces for portlets to cleanly separate the portlet from the portal infrastructure. For the most part, the portlet API is an extension of the servlet API, except that it restricts certain functions to a subset that makes sense for portlets running in the context of a portal. For example, unlike servlets, portlets cannot send errors or redirect messages as a response. This can be performed only by the portal itself, which controls the overall response page.

The markup fragments that portlets produce can contain links, actions and other content. The portlet API defines URL rewriting methods that allow portlets to transparently create links without needing to know how URLs are structured in the particular portal.

Portlet performance

Because portlets are essentially servlets, similar reentrance and performance considerations apply to both. A single portlet instance (that is, a single instance of the portlet's Java class) is shared among all requesters. A limited number of threads can process portlets and servlets, so each portlet must do its job as quickly as possible to optimize response time for the whole page. Just as with servlet programming, you should consider optimizations such as limiting the use of synchronized methods, limiting the use of expensive string operations, avoiding long-running loops and minimizing the number of objects created. You can also optimize response times by using JavaServer Pages (JSP) to render the portlet's views. In general, views created with JSP are faster than views created with Extensible Stylesheet Language (XSL).

Usually, several portlets are invoked in the course of handling a single request, each one appending its content to the overall page. Some portlets can be rendered in parallel, so that the portal server assembles all the markup fragments when all the portlets finish or time out. This improves the performance of portlets that access remote data by HTTP or Simple Object Access Protocol (SOAP) requests. However, not all portlets are thread-safe. For example, portlets that access protected resources cannot run in parallel. The portlet deployment descriptor indicates whether the portlet is considered thread-safe. Portlets that are not thread-safe are rendered sequentially.

Portlet output can also be cached. Caching policies are configured in the portlet deployment descriptor. You can include an expiration time and whether the portlet markup can be shared among users or is user-specific.

Standards

As portals continue to evolve into the new desktop and integration standard, IBM directs efforts to standardize the APIs between portals and other applications, and often assumes lead technical positions within many standards organizations. In particular, the Java Community Process (JCP) and the Organization for the Advancement of Structured Information Standards (OASIS) work cooperatively to standardize the Java and XML technology needed to link portals to disparate applications. Some of these standards include or will include:

- *Java Specification Request (JSR) 168, a specification from JCP that addresses the requirements of aggregation, personalization, presentation and security for portlets running in a portal environment. JSR 168 is co-led by IBM and Sun Microsystems, Inc., and is designed to facilitate interoperability between local portlets and portal servers. WebSphere Portal includes a portlet run-time environment with a JSR 168 portlet container that supports operation of portlets developed according to the Java Portlet Specification defined by JSR 168.*
- *JSR 170, a proposed specification from JCP designed to implement a standard meta-model definition and an API for bidirectional, granular access to content repositories. This should result in platform-independent methods to interact across content-management solutions and associated services including versioning, search, content categorization, access control and event monitoring. IBM is a participant in the expert-member group defining the JSR 170 content repository for Java technology APIs. When JSR 170 is published, IBM plans to support this standard across its content-management offerings, and will use the JSR 170 repository to store all portal content.*
- *OASIS Web Services for Remote Portals (WSRP), an XML and Web services standard that enables the interoperability of visual, user-facing services with portals or other Web applications. The OASIS WSRP standard simplifies the integration of remote applications and content into portals by defining a Web service interface for interactive presentation-oriented Web services. The producer part of the WSRP implementation provides an entry point into the producer portal, enabling the portal to provide portlet applications or single portlets as WSRP services. A WSRP consumer is a portal that wants to integrate WSRP services and consume them as remote portlets from portals that provide them. As a result, using WSRP makes integrating content and applications into portals easier, eliminating the requirement for custom coding or the use of a variety of protocols and interfaces to adapt Web services for use in their portal implementation. WebSphere Portal, Version 5.1 includes support for producer and consumer WSRP services.*

- *Struts, a Jakarta open-source project that provides a framework based on the MVC pattern. It enables developers to efficiently separate an application's business logic from its presentation. Struts enforces sequences of pages and actions and provides form-validation functions. WebSphere Portal, Version 5.1 includes support for the Struts, Version 1.1 framework to build portlets. To work in portlets, you must observe specific technical details when using Struts. For example, when used in portlets, a Struts action should not write to the response object, and should not create header elements like HEAD and BODY tags. The Struts application must be packaged with several replacement Java Archive (JAR) files that help ensure that URLs, forward actions and include actions run correctly in a portal environment.*

Portlet cooperation

The portal server provides a mechanism for portlets to communicate with each other, exchanging data or other messages. In a production portal, you can use portlet communication to copy common data between portlets. This capability saves time by minimizing the need for redundant typing by the user and makes the portal easier to use. For example, one portlet might display information about accounts while a second portlet displays information about transactions that have occurred for one of the accounts over the past 30 days. To do this, the transactions portlet needs to obtain the corresponding account information when it displays the transaction details. This action is accomplished by communication between the two portlets, using portlet actions and portlet messages. For example, the account portlet creates a portlet action and encodes it into the URL that is rendered for displaying transactions. When the link is clicked, the action listener is called, and a portlet message is launched to send the necessary data.

Programmatic messaging helps unify portlet applications that access different back-end applications. However, programmatic messaging is relatively static, and requires planning and design work in advance. The portlets that are exchanging messages must already know about each other to make the interchange work. The next section discusses a more flexible means of portlet cooperation.

Brokered cooperation

Brokered cooperation enables independently developed portlets to exchange information. Portlets register their intent to cooperate with a broker, which facilitates the exchanges at run time. The broker works by matching data types between the sources in one portlet and the actions of another portlet. When the types match, a transfer is possible, and the broker enables the user to trigger the transfer through a pop-up menu. The term *click-to-action* is used to describe this menu-driven, brokered data exchange (see Figure 5).

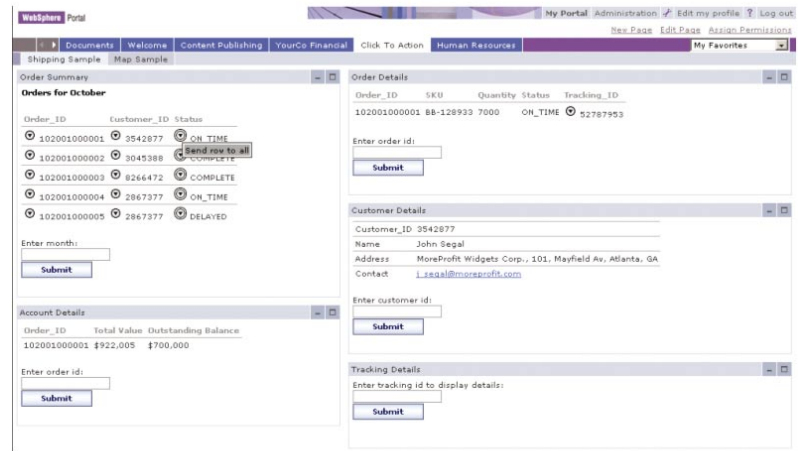


Figure 5. Exchanging information through the click-to-action feature

The objective of click-to-action portlets is to increase the productivity of portal users working with multiple portlets by easily enabling them to send information from one portlet to another. For example, users can click information that is displayed in one portlet and transfer that information to another portlet. The portlet receiving the information processes it and updates its display. The click-to-action capability automatically matches portlet information sources and possible actions based on their data-type compatibility. Click-to-action does not rely on drag-and-drop or other nonstandard browser features. It also offers the unique advantage of being able to work in different browsers – making it more accessible to users.

An extension of this idea, *cooperative portlets*, is also supported by WebSphere Portal. Using cooperative-portlet capabilities, administrators can prewire portlets so that they exchange data automatically (see Figure 6). Data transfers along the wires using the same broker as click-to-action. Besides saving the extra step of having the user click the data source to select a target, wiring portlets together enables greater flexibility to match brokered values.

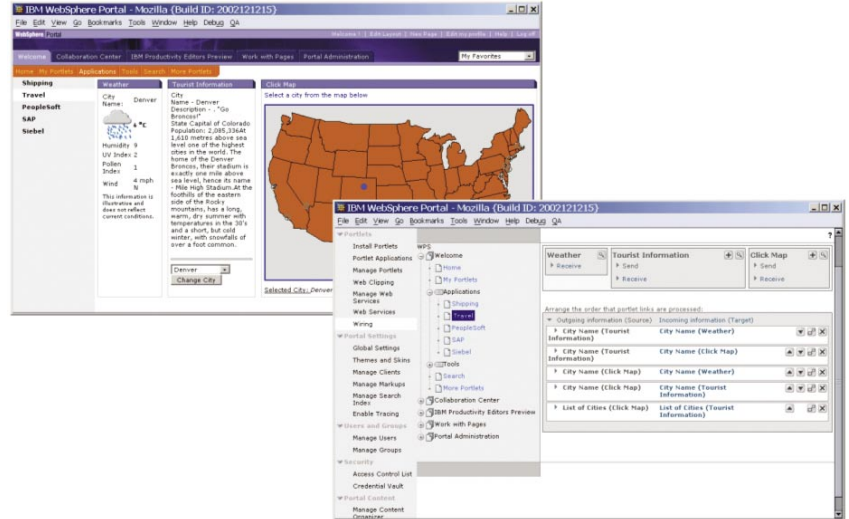


Figure 6. Prewired cooperating portlets

Discoverable services

The portlet API provides an interface to enable dynamic discovery of available services. Each service is registered in a portal configuration file, and is accessed from the `PortletContext.getService()` method, which searches the factory for the service, creates the service and returns it to the portlet. This method makes services available to all portlets without having to package the service code with the portlet. And you can exchange or enhance the implementation of this kind of service transparently – without affecting the portlet.

The portal server provides discoverable services for its credential vault to manage persistent TCP/IP connections and to manage the portal content repository. Portal developers can implement new services as options, to further customize their infrastructure, such as location, notification, content access or mail services.

Development tools

WebSphere Portal includes a range of development options, from nonprogrammatic business-user tools, Web clipping and Web services application-integration techniques, to Java 2 Platform, Enterprise Edition (J2EE) technology-based portlet and portlet development tools to provide interactive access and data cooperation services across a range of Web, database, content management, enterprise resource planning (ERP), customer relationship management (CRM) and other solutions. More developer tools supporting portlet and portal development are available from IBM Business Partners.

IBM Rational Application Developer, Version 6.0 is a comprehensive integrated development environment (IDE), with full support for the J2EE programming model, including Web, Java, Web services, EJB and portal application development (see Figure 7). The product includes a set of visual portal development tools and a WebSphere Portal test environment, enabling you to build and test individual portlets and entire portal applications. Portlet wizards create a complete portlet that complies with the IBM Portlet API, as well as the new JSR 168 Portlet API, the industry-standard specification that addresses the requirements of aggregation, personalization, presentation and security for portlets running in a portal environment.

The visual portlet-development tool enables you to build rich user interfaces for portlets quickly with JavaServer Faces (JSF) components that generate code for event handling, user-input validation and data handling. These tools also connect portlets to relational databases, Enterprise JavaBeans (EJB) components, Web services and enterprise information systems (EISs), such as SAP and Siebel, through point-and-click operations (see Figure 8).

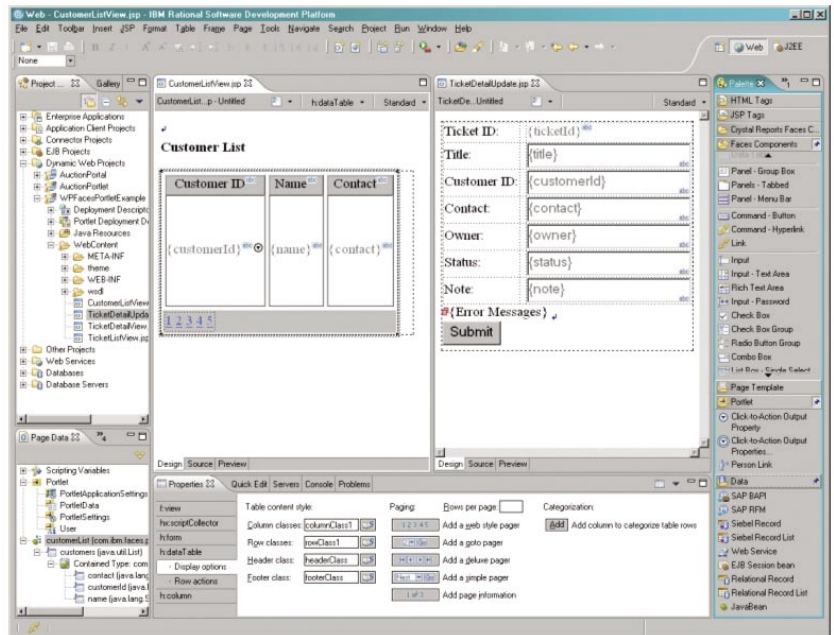


Figure 7. Visual portlet development using Rational Application Developer

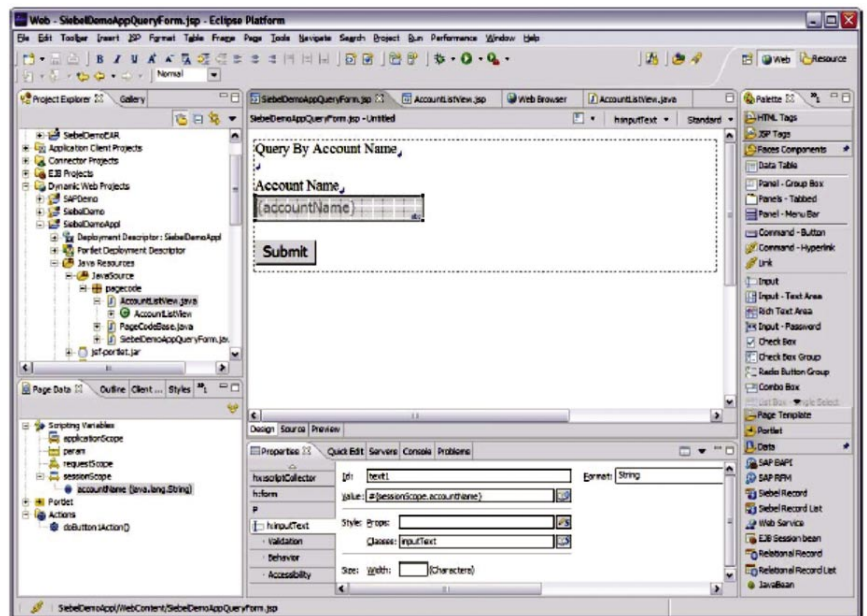


Figure 8. Using Rational Application Developer to build portlets to access EISs

Rational Application Developer also provides the visual portal site development tools. The portal designer enables you to create portal pages and customize the layout of portlets, and edit portal themes and skins that control portal site appearance. The created portal site can be tested on WebSphere Portal test environment or on a separate portal server (see Figure 9).

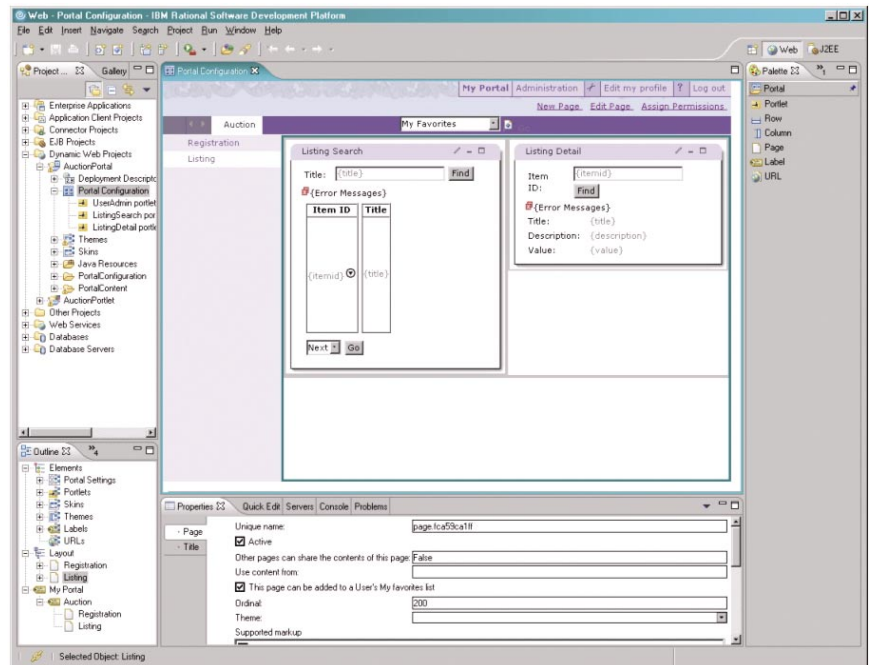


Figure 9. Customizing a portal site using the portal designer feature in Rational Application Developer

Content management

When companies deploy portals, they want to view, organize, share and find information from various sources, in various formats. WebSphere Portal has comprehensive content-management capabilities, providing support for syndicated content, document management and Web content management. It can integrate with leading Web content-management systems, and includes advanced personalization capabilities and integrated search services, portlets, and features that deliver categorization and summarization functions.

Syndicated content

A key concept related to portal technology is *syndication*, which means the ability to deliver fresh, personalized, and filtered content and services from multiple sources to subscribers. Typically, the content relates to news, finance and entertainment (see Figure 10). Companies that team with IBM to provide this content include MarketWatch, Dialog, Moreover, YellowBrix, Hoovers, Factiva, NewsEdge, MediaApps and DataMonitor.

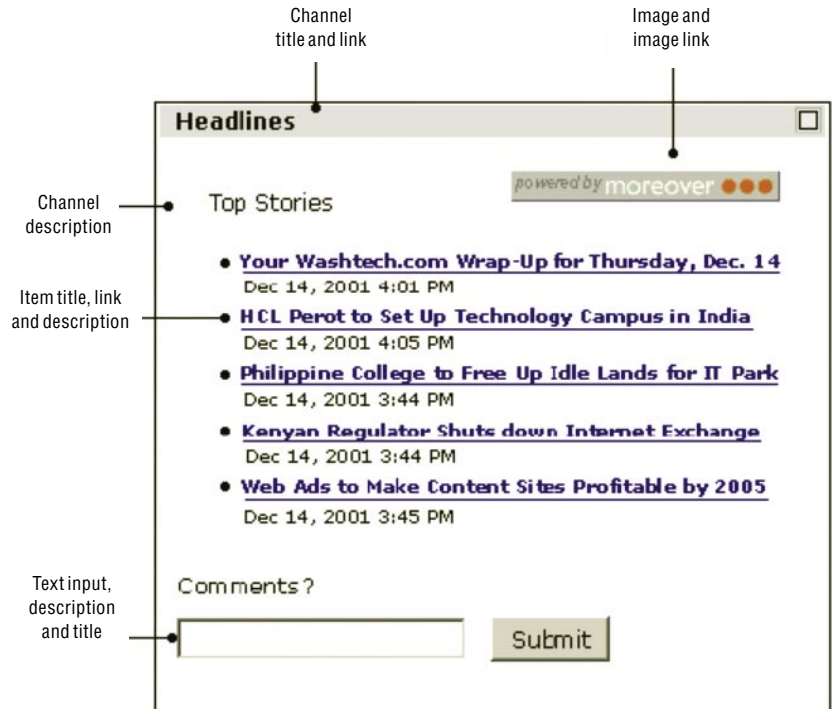


Figure 10. WebSphere Portal supports standards-based content integration and display.

Companies are embracing syndication concepts and standards to automate the publishing of electronic catalogs and other internal information, and to make this information available to users through enterprise portals. A popular and useful format for syndicated news and entertainment content is really simple syndication (RSS). Content can be published directly from the content-management system into RSS and Open Content Syndication (OCS) channels, where the portal server's built-in RSS portlet can easily display the content. This self-syndication concept defines a procedure for editing, managing and publishing your own sources of content.

Enterprise content management

To manage and search large collections of content – spanning multiple content sources, including databases, records, images, reports, electronic statements, audio, video and Web content – created and used by a variety of applications, you need an enterprise content-management solution like IBM Content Manager. IBM Content Manager provides a portlet-based interface that enables you to leverage its enterprise content-management features within the WebSphere Portal environment. Portlets enabling integration with other leading content-management and search vendors (like Documentum, Fatwire, Vignette, Stellent, Autonomy, Fast and Verity) are also available and can be found in the WebSphere Portal Catalog.

Document management

WebSphere Portal includes document-management capabilities, designed to provide an efficient, centralized repository and management service that enables teams to work collaboratively to author, organize and share documents and other content. The WebSphere Portal Document Manager allows authorized users to view, add, edit, navigate and search across documents within a user-defined folder hierarchy, as shown in Figure 11.

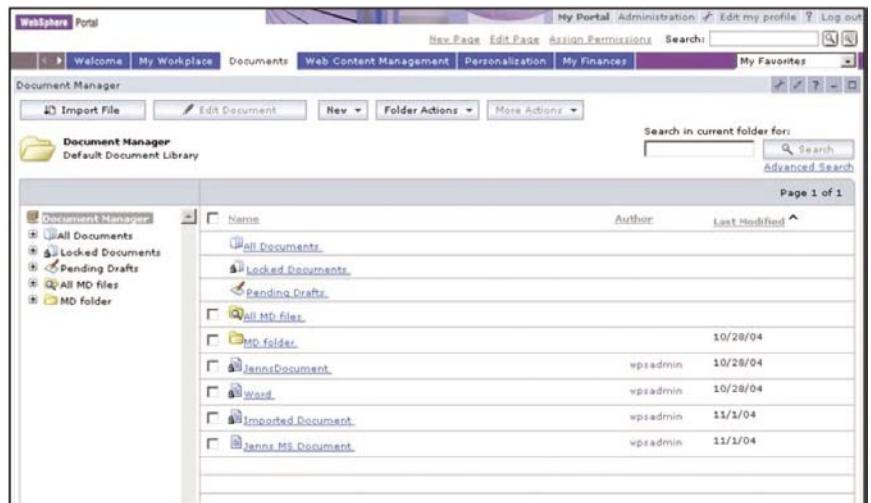


Figure 11. Portal document management

You can make new and changed documents available immediately or manage them through versioning and approval processes. You can forward document links and folder links through e-mail, enabling users to receive notifications of new and updated content. Documents produced using WebSphere Portal Document Manager can be accessed and published by IBM® Workplace Web Content Management™. These documents can also be the subject of business rules developed with the WebSphere Portal personalization component. And WebSphere Portal Document Manager software can leverage Lotus Instant Messaging services by surfacing integrated presence awareness and chat services to support team content contributors.

WebSphere Portal Document Manager uses WebSphere Portal access-control capabilities to regulate which users can view or edit specific documents. Privileges can be set on a document library, project, folder, subfolder or individual document. All documents or subfolders inherit the privileges set on a folder contained in that document, unless specifically blocked. This capability makes it easy to efficiently manage access rights on a large collection of documents.

By integrating with the WebSphere Portal search-engine component, WebSphere Portal Document Manager enables users to search on document content, title, description or author, as shown in Figure 12. Results are returned with relevance ratings to the requested search criteria. Documents can include text documents such as HTML, or application-specific documents such as Microsoft Word. Users can view Microsoft Word and more than 250 file formats from WebSphere Portal Document Manager using embedded file viewers. Users can also invoke workstation applications, such as Microsoft Word, directly from the WebSphere Portal Document Manager portlet interface, providing a seamless way to create and edit documents without having to leave the portal user experience.

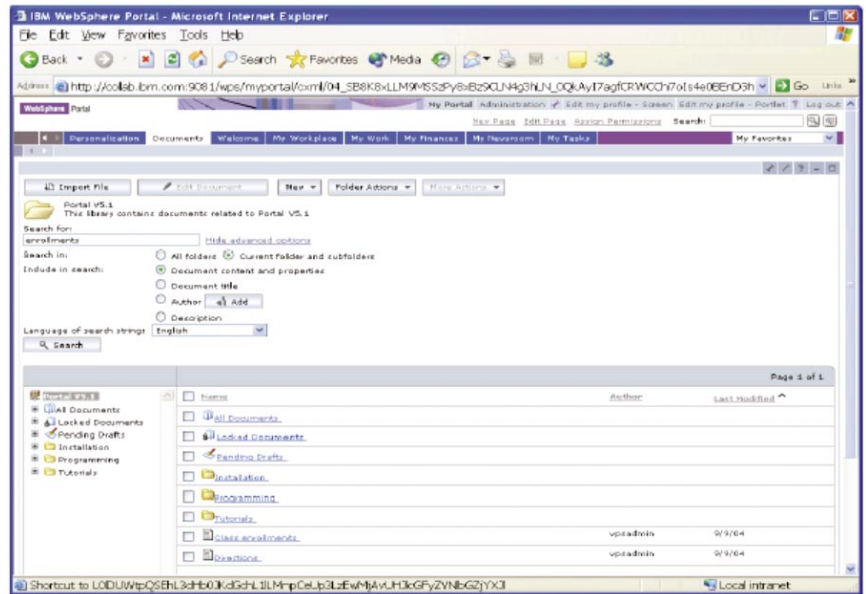


Figure 12. Advance search features in WebSphere Portal Document Manager

Web content management

Web content-management capabilities enable users to create, approve and publish Web content to Web servers. The steps of this process include defining content types, roles, publication options, destination specifications and workflow processes. Many content-management products are in the marketplace today, including IBM® Workplace Web Content Management™, and other IBM and vendor content-management products. Although each of these products works differently, generally, they are designed to create, maintain and publish collections of structured and unstructured content that can be made available to users through a portal.

WebSphere Portal includes integration kits (available through WebSphere Portal Catalog) that illustrate the specific steps required to publish RSS content from several of the previously mentioned Web content-management products. Content contribution and approval operations of the Web content-management system can also be accessed through portlets provided by each respective company. These portlets provide a user interface into various aspects of the content-management process, such as content submission, workflow management, content approval, and even staging or publishing.

WebSphere Portal includes IBM® Workplace Web Content Management™ software, which enables users – such as business users creating template-driven content like press releases or product information, or graphic artists creating and editing artwork using their preferred tools – to contribute content to a Web site as their roles dictate. IBM® Workplace Web Content Management™ supports content contribution through templates or forms and provides specific support to publish content maintained in the WebSphere Portal Document Manager repository. It also supports the ability to contribute files, such as images, HTML or JSP that are created and edited using popular tools such as Macromedia Dreamweaver, Microsoft FrontPage, Adobe Photoshop or Microsoft Word.

Through IBM® Workplace Web Content Management™, users can create elements of a Web page as distinct units, and then put them together in a variety of ways to support componentization and reuse of content resources. This capability yields flexible Web pages that are simple for authors to create and easy for site administrators to maintain (see Figure 13). The product’s portlet interface enables users to contribute content to a Web site in an easy-to-use, quick and controlled manner. The concept of componentization remains consistent throughout authoring, site design and maintenance processes. This consistency enables large teams to work together – jointly when applicable, and individually to manage specific tasks.

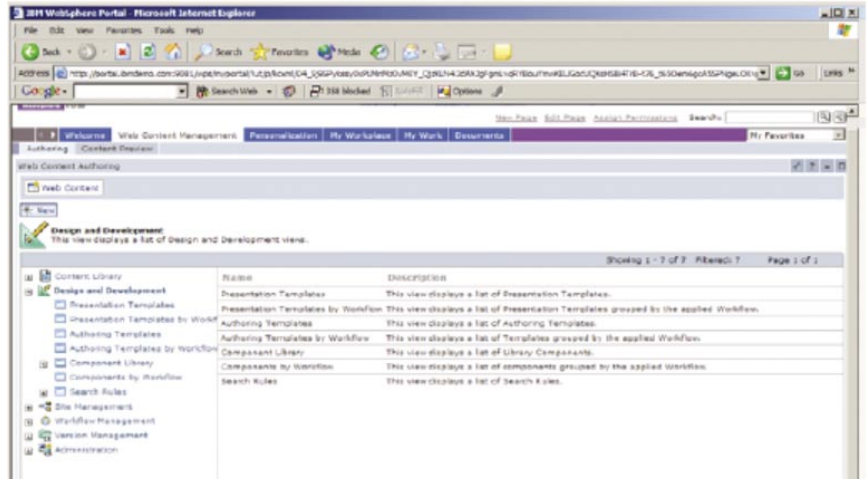


Figure 13. IBM® Workplace Web Content Management™ authoring features

You can use IBM® Workplace Web Content Management™ content-publishing tools to manage content that is served through portlets within the portal server. Designated contributors and administrators can organize the content and design of pages in a Web site, framework and navigation of site content, and supporting content-management services such as edits and approvals, caching and staging services, and publication of Web content to the portal. IBM® Workplace Web Content Management™ software includes support to author personalization rules and campaigns to target content to specific groups or users and to report on the content usage within your portal. The product also provides workflow approval options to manage content publishing to help you maintain access control over what content each user can see or change, and coordinates the approval and publishing process when the content is ready (see Figure 14).

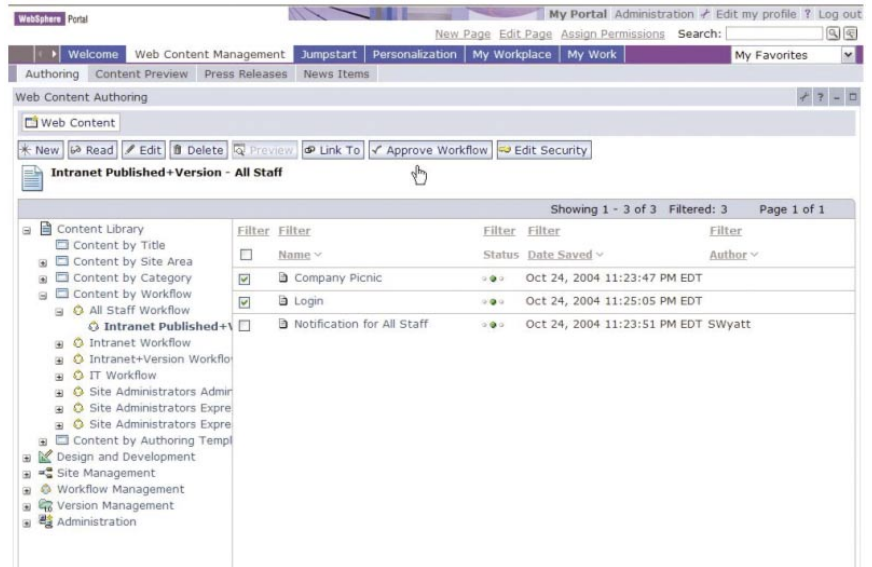


Figure 14. IBM® Workplace Web Content Management™ workflow and content-publishing approvals

Productivity components

WebSphere Portal also includes productivity components, portal-based content edit capabilities and conversion services. These productivity components can be accessed from WebSphere Portal Document Manager, IBM® Workplace Web Content Management™ and the IBM Common PIM portals. Productivity components function as embedded editors for rich text, spreadsheets and presentations, providing the most widely used content-editing features – without your having to install office-productivity software on every workstation. Instead, these editors enable users to create and edit document types from within their Web browsers, as shown in Figure 15.

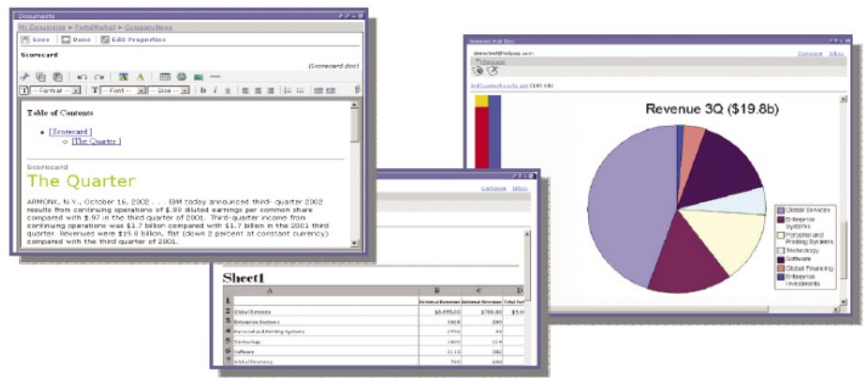


Figure 15. Create, view and edit content in WebSphere Portal using productivity components.

Content from popular office-productivity products can be opened for edit from WebSphere Portal productivity components, and content generated by the editors can be converted to supported formats, including IBM Lotus SmartSuite®, Microsoft Excel and Microsoft Word documents. And document-conversion services included with WebSphere Portal support view, index and search capabilities for over 250 file types.

Search

WebSphere Portal provides integrated Web-content indexing capabilities, as well as methods to categorize indexed content and an optional workflow-approvals process to manage publication of indexed content. WebSphere Portal provides two sets of search portlets. A document search and browse portlet can be deployed on a portal page to allow users to search within one indexed search collection. It also enables users to access advanced capabilities such as browse and search by optional applied taxonomy of indexed content, and the ability to conduct advanced searches using fielded search constraints, such as search by document type, author and date range. A search box is also provided in the portal interface, and includes sample themes for customization of appearance. Users can enter a search term in the search box, which navigates the user to the search-center portlet. This portlet enables centralized searches across WebSphere Portal content, including portlets and portal pages, WebSphere Portal Document Manager content, Web-indexed applications and external Web source sites. The embedded search service uses WebSphere Portal document-conversion services.

Besides the search capabilities available through document-conversion services, the WebSphere Portal built-in search engine is optimized to support full-text searching of small-to-midsize business content collections where precision is essential. It efficiently applies state-of-the-art search algorithms to produce high-quality search results. You can index the content, allowing users to search for content that is accessible through HTTP.

The search engine supports free-text queries, with query assistance and query word completion, and also supports wildcard and fielded search options. Search queries can also use advanced query operators (+ or -) to indicate keywords that must be or also must not be in the document. Users can also search for explicit phrases by enclosing them in double quotation marks. The search engine can search documents in virtually any language, and also supports synonyms and stop-word lists during query processing. Search results include document summaries, category information (if applied) and other standard attributes (date, URL, author and relevance ranking). See Figure 16 for an example of the taxonomy browse search feature.

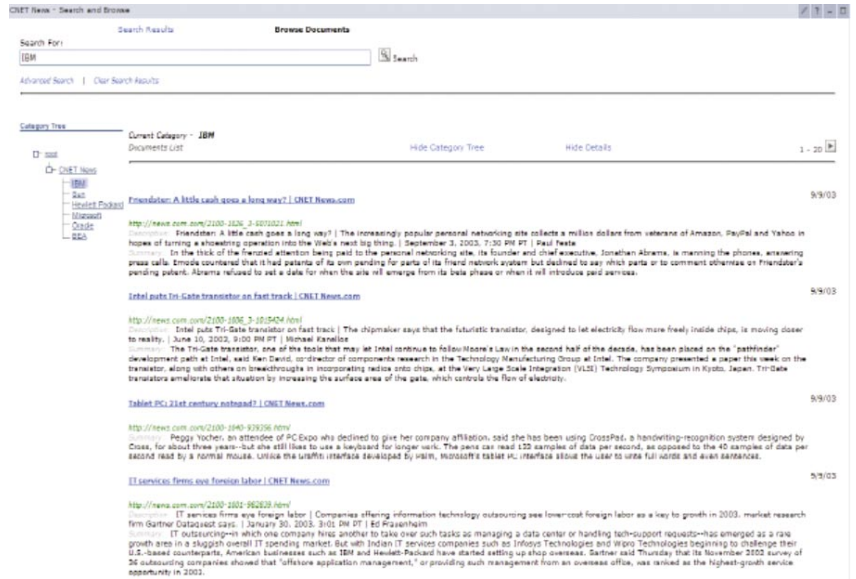


Figure 16. Navigating taxonomy categories to search content using the document-search portlet

To prepare for a search, the search engine builds a full-text index from content it collects from the local portal site, document and Web site sources. The index function supports multiword and international-language content indexing to avoid ambiguity and to help ensure high precision. You can compress these indexes and control their sizes to accommodate situations in which the size of an index needs to be limited. Administration portlets enable you to create, update and manage index parameters and optional categorization options (see Figure 17). Content-categorization options include classification to preestablished categories or a rules-based method to define content categorization filters to apply to indexed content. Administrators can choose to configure specific portlets provided with WebSphere Portal to view and edit the taxonomy characteristics before applying them to evaluate content that is indexed by the WebSphere Portal search engine.

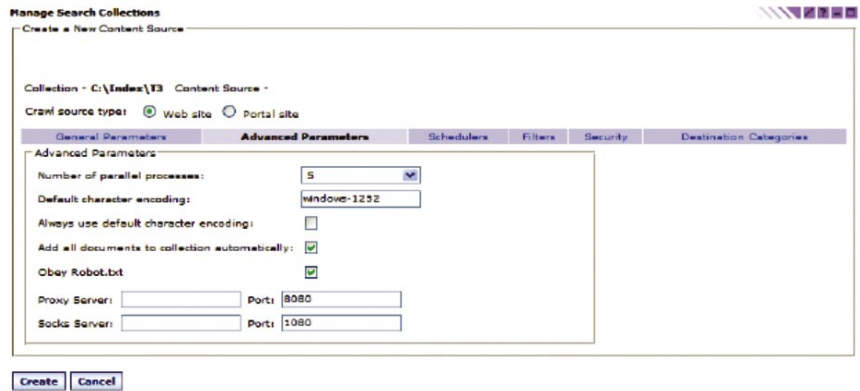


Figure 17. Configuring the search index

WebSphere Portal also integrates several other search technologies. Portlets using Lotus Extended Search and IBM WebSphere Information Integrator: OmniFind Edition can access and aggregate other search engines and indexes in a distributed fashion. If you need to offer heterogeneous search capabilities across a broad range of data, Web sites and indexed-content sources to complement the WebSphere Portal search engine embedded search services, you should consider using Lotus Extended Search (included with the WebSphere Portal Extend offering). If you need to provide enterprise-scale index and search services across a broad range of Web, data and content sources, you should consider WebSphere Information Integrator: OmniFind software.

Lotus Extended Search

Lotus Extended Search provides distributed, heterogeneous searching across a broad range of IBM Lotus Domino, database and Internet sources to return aggregated, ranked results to a single interface (see Figure 18). The result is single point of access to a variety of data sources without requiring a new, central index. Lotus Extended Search software can search and retrieve documents from repositories that include IBM Lotus Notes, Lotus Team Workplace, IBM Lotus Domino Document Manager and IBM Lotus Domino Server Domain Index.

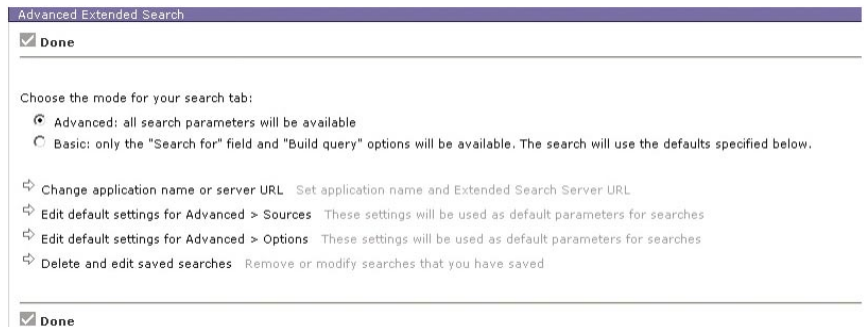


Figure 18. Configuring the Lotus Extended Search portlet

Lotus Extended Search also searches external sources such as Microsoft Index Server and Microsoft Site Server, Microsoft SharePoint, WebSphere Portal search-engine indexes, LDAP technology-compliant directories, 18 popular Web search sites and news sites, commercial content providers, and relational databases such as IBM DB2[®], Oracle, Microsoft SQL Server and other Open Database Connectivity (ODBC) technology-compliant databases. Results can be ranked by relevancy from the source, across multiple data stores. Lotus Extended Search portlets provide basic and advanced search options that enable end users to select across supported sources and refine search criteria.

WebSphere Information Integrator: OmniFind

WebSphere Information Integrator: OmniFind, available separately, is designed to provide enterprise-scale index and search services across a broad range of Web, data and content sources. Supported sources include HTTP and HTTP Secure Sockets (HTTPS); news groups (using Network News Transfer Protocol [NNTP]); file systems; Lotus Domino databases; Microsoft Exchange public folders; IBM DB2 Content Manager, IBM DB2[®] Universal Database[™], IBM DB2 Universal Database for z/OS[®], IBM Informix[®] and Oracle databases; and a data listener and API to facilitate integration with other content search sources. WebSphere Information Integrator: OmniFind includes sophisticated text-analysis and relevancy algorithms, and the ability to index millions of documents. This product also typically provides subsecond search-response times. Advanced search options include predefined link customizations, parametric and fielded search options, and content categorization and taxonomy management services. WebSphere Information Integrator: OmniFind also includes administrative portlets and search-server performance-monitoring features. If you require enterprise-scale index and search services, more-extensive search options and broader support of search sources than those supported with WebSphere Portal search-engine services, use WebSphere Information Integrator: OmniFind. Its migration utility enables you easily grow your search infrastructure from WebSphere Portal search-engine embedded services to more-scalable WebSphere Information Integrator.

Security

With the explosive growth of B2B and B2C Web applications, an on demand business needs to protect critical information assets. Service providers need similar protection when rehosting Web-based content and applications for their customers. Portal applications and resources are protected by:

- *Managing user profiles (member services)*
- *Verifying user identity (authentication)*
- *Managing access to back-end applications (single signon)*
- *Enforcing access policies (authorization)*

The next section highlights some of the security features that applications and portal administrators can use to better protect their portals' valuable information assets.

Member services

Centralized administration of user identities, credentials and permissions is desirable in many environments. WebSphere Portal includes facilities to define portal users and manage user-access rights. The user and group subsystem includes portlets that enable users to register and manage their own account information, administration portlets to manage user accounts and group information, and a repository that stores all the information about portal users (see Figure 19). The subsystem also provides services to create, read, update, and delete users or groups in the repository. User profile information includes general information such as a user's name and user ID, as well as preference information such as news topics of interest and preferred language. A user can be a member of one or more groups, and groups can contain other groups.

Dialog box titled "Edit My Profile" with a help icon, close icon, and maximize icon. The text inside reads: "New user enrollment. Provide the information requested below and click OK. Click Cancel if you do not want to sign up at this time." Below this are several input fields, each preceded by an asterisk: "* User ID:", "* Password:", "* Confirm Password:", "* First Name:", "* Last Name:", "Email:", "Preferred language:". The "Preferred language:" field is a dropdown menu currently showing "- Nothing Selected -". At the bottom, there is a legend: "* Required Field" and two buttons: "OK" and "Cancel".

Figure 19. Self-enrollment page

The default user profile attributes are based on the inetOrgPerson schema, as shown in Figure 20, which is supported by most LDAP directories. The user repository can consist of multiple data sources. By default, the repository includes two data sources – a combination of a database and a directory server. The database can be any of the databases supported by WebSphere Portal. Any one of several LDAP directory products are supported, including Netscape (iPlanet) Directory Server, Microsoft Active Directory, Novell eDirectory, Lotus Domino and IBM Tivoli Directory Server.

```
<MemberServiceProperties>
  <Database ... DatasourceName="@DATA_SOURCE@" />
  ...
  <Directory userRDNname="uid"
    userMemberSubsystemAttributeName="loginId"
    userObjectClass="top;inetOrgPerson"
    ...
    orgRDNname="dc"
    orgMemberSubsystemAttributeName="orgEntityName"
    orgObjectClass="top;domain"
    ...
    orgUnitRDNname="cn"
    orgUnitMemberSubsystemAttributeName="orgEntityName"
    orgUnitObjectClass="top;container"
    ...
    grpRDNname="cn"
    grpMemberSubsystemAttributeName="memberGroupName"
    grpObjectClass="top;groupOfUniqueNames"
    grpMembershipAttributeName="uniqueMember"
    ...
  </Directory>
</MemberServiceProperties>
```

Figure 20. Mapping user profile data defined to LDAP schema

The *realm* concept, newly supported with WebSphere Portal, Version 5.1, provides a more-flexible method to define an individual user population for the virtual-portal feature. This concept enables physical user-registry systems to be abstracted. The resulting realms expose a distributed user population, for instance, identified in different trees of an LDAP structure, as a single coherent entity to the specific virtual portal instance. The mapping of a realm to a virtual portal defines the individual user population that can work with a specific virtual portal. Although it is possible to aggregate different user repositories into one realm, WebSphere Portal, Version 5.1 only supports aggregating different repository trees of one LDAP user registry. The configuration for the user repositories to be used, the realm setup, and the definitions of the user profile attributes and their mappings to LDAP object classes is handled in the member-management XML configuration files.

The mapping of user profile attributes to LDAP object classes is defined by using the `wms.xml` file. This file specifies the names of the various data repositories and how they are navigated to retrieve user and group information. You can configure these settings differently for each supported LDAP directory. If you want to use a directory that is not supported, you would need to set these values appropriately for that directory server. The `attributeMap.xml` file specifies the details of how each attribute is mapped to the LDAP directory or database. This mapping file also includes metadata about each attribute such as its data type, whether it is required, whether it can have multiple values and so on.

Administration

Administration of users and groups can be performed by users themselves (called *self-care*) or by portal administrators. The portal server includes portlets to register new users as well as administration portlets to update user and group information. You can easily modify the registration and self-care portlets to accommodate new attributes. You can simply switch to the configure mode of the portlet and modify the set of attributes. The enrollment servlet automatically stores the new data in the corresponding user attributes. The WebSphere Portal information center, available at ibm.com/developerworks/websphere/zones/portal/proddoc.html, includes more information about customizing the implementation of the user repository, registration and self-care portlets, and data-validation classes.

Authentication

Authentication is the process of establishing a user's identity. Usually, the portal server uses the authentication services provided by WebSphere Application Server. You can also opt to use another authentication server (such as IBM Tivoli Access Manager WebSeal or Netegrity SiteMinder) that has a trusted association with the application server. WebSphere Portal software uses form-based authentication. This authentication method means that a user is prompted through an HTML form for the user ID and password for authentication when trying to access the portal. WebSphere Portal, Version 5.1 updates the login form provided with earlier releases with a new, easy-to-use login portlet. The portal server requests the application server to validate the authentication information against an LDAP user registry. WebSphere Portal, Version 5.1 also now supports login using Secure Sockets Layer (SSL) client certificates.

WebSphere Application Server uses Lightweight Third Party Authentication (LTPA) as its authentication mechanism. When a user tries to access a protected resource, the application server intercepts the request and redirects the request to the login form, if form-based login is activated. This form posts the user ID and password to the portal, which triggers a Java Authentication and Authorization Service (JAAS) login for a certain JAAS login configuration against the application server to authenticate the user. If the user can be authenticated, the security context is established by building a JAAS subject, and an LPTA cookie is stored on the user's machine. The JAAS subject groups the authenticated user's security-related information in the form of principals (such as the groups the user is member of) and credentials (such as public key certificates).

If your system uses another vendor's authentication server, trust needs to be established between that proxy and WebSphere Application Server. This trust is accomplished using a Trust Association Interceptor (TAI) module, which converts security information specific to the authentication proxy into a format that can be handled by the application server. The supported authentication mechanism depends on the capabilities of the other vendor's product.

When a user tries to access the portal, the authentication proxy from the other vendor's product intercepts the request and challenges the user to authenticate. After a successful login, the original user request, along with other security information in the request header, is passed to the application server.

The format and content of this information is vendor-specific. WebSphere Application Server uses the TAI module (specific to the other vendor's product) to extract the necessary security information from the request header. TAI modules for Tivoli Access Manager and Netegrity SiteMinder are packaged with the portal server (all editions). The WebSphere Application Server information center, available at ibm.com/software/webservers/appserv/was/library/, includes information about creating custom TAI modules for other vendors' reverse-proxy servers.

Many portlets need to access remote applications that require some form of user authentication. For accessing applications outside the portal's realm, WebSphere Portal provides a credential vault service that portlets can use to store user IDs and passwords (or other credentials) for a user login to an application (see Figure 21). Portlets can use these services on behalf of the user to access remote systems. The credential vault supports either local database storage or Tivoli Access Manager for secure storage and retrieval of credentials.

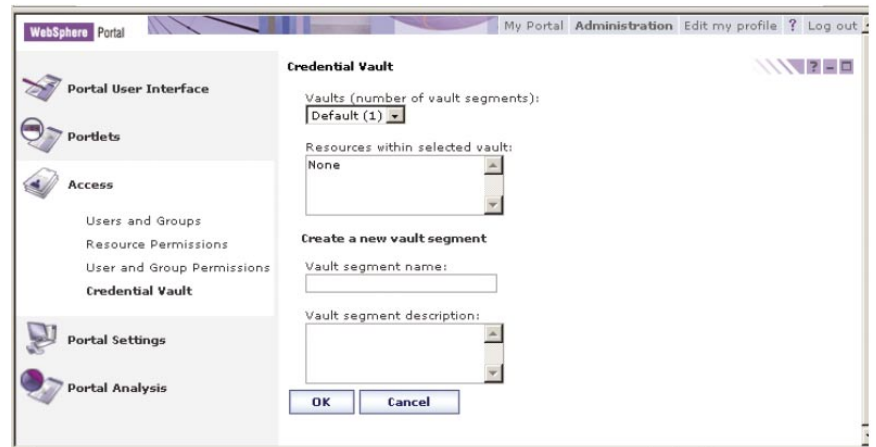


Figure 21. Setting up the credential vault

Portlets access credentials by obtaining a `CredentialVaultPortletService` object and calling its `getCredential` method. The returned credential includes two options:

- *It can use passwords or keys from a passive credential, passing them in application-specific calls. Portlets that use passive credentials need to extract the secret out of the credential and do all the authentication communication with the back-end application.*
- *It can call the `authenticate` method of an active credential. Active credential objects hide the credential's secret from the portlet, with no way to extract it from the credential. Active credentials provide other methods to perform the authentication.*

The latter option allows portlets to trigger authentication to remote servers using basic authentication, SSL client authentication, digest authentication or LTPA without knowing the credential values. Using active credentials means that the portal authenticates on behalf of the portlet, and the portlet can simply use the open connection. Some remote applications use form-based logins and store cookies during the login-form processing. You can use the `HttpFormBasedCredential` object to handle these form-based logins and to store all the cookies that are returned as a result. For subsequent calls, the portlet can then ask the credential for an authenticated connection. This provides an HTTP connection with these cookies already set in the header, enabling portlets to maintain persistent, secure back-end connections with active credentials. Although using the active-credentials option might not be possible in all cases, it is the preferred technique. For highly secure transmission of data, portlets can request a secure session (using HTTPS) for accessing Web applications.

WebSphere Portal uses the JAAS architecture. JAAS provides a means to authenticate subjects and to provide fine-grained access control. JAAS is part of the standard Java security model; it gives applications independence from the underlying authentication and authorization mechanisms being used. In a distributed J2EE environment, portlets can use the JAAS API to access JAAS technology-enabled back-end applications.

WebSphere Portal also supports enablement of Java 2 Standard Edition (J2SE) security within the WebSphere Portal environment. J2SE provides extended access-control mechanisms that increase overall system integrity by checking for permissions before allowing access to certain protected system resources. J2SE security allows you to set up individual policy files that control the privileges assigned to individual code sources, including access to specific Java Virtual Machine (JVM) environments, server nodes, libraries and applications.

The WebSphere Portal, Version 5.1 installation procedure grants all code shipping with WebSphere Portal full access to system resources. All other code running within the portal environment runs with permissions granted according to the WebSphere Java 2 Platform security setup. If a portlet needs other permissions to operate correctly (such as to write report data to the file system), a policy file can be inserted into the corresponding WAR file. The administrator should check that the policy file adheres to the system's security policy, and if necessary, modify the permissions before installing the portlet.

Single signon

Having to sign on multiple times just to access different portal applications can frustrate users and slow their progress. Users want to be able to sign on only once and have access to the different parts of the portal server with the same consistent user credentials. To address this need, WebSphere Portal delivers comprehensive single-signon support by using WebSphere Application Server and authentication proxies. This means that WebSphere Portal delivers the capabilities to enable users to log on only once to gain access to all the enterprise applications you have installed within the single-signon realm.

WebSphere Application Server uses LTPA tokens to provide single-signon capabilities. When a user is authenticated, the portal server creates an LTPA single-signon cookie containing the authenticated user credential. This encrypted cookie conforms to the format used by WebSphere Application Server, and all application servers in the shared domain can decrypt this cookie if they all have the same cipher key. Through the single-signon cookie, all servers in the cluster can access the user's credentials without more prompting, resulting in a seamless access experience for the user. To benefit from the LTPA single sign-on method, users must have their Web browsers set to support cookies and enable support for session cookies.

Authorization

After determining the user’s identity, the portal server consults locally cached access-control lists to determine which pages and portlets a user has permission to access. WebSphere Portal enforces access control to portal assets, including portlets, pages and user groups. The portal administration database stores the access control lists. You can also manage access control for specific resources in an external security manager, such as Tivoli Access Manager or Netegrity SiteMinder.

Access permissions are maintained using the access-control administration portlet (see Figure 22 and Figure 23). You can use this portlet to assign roles to individual users or to groups of users for specific portlets, pages or documents. Roles are permission sets, such as the ability to view and update the corresponding item. Users can also delegate the permissions they hold to other users. When a role is assigned to a user or a group on a container (such as a page that contains portlets or other pages, or a folder that contains other folders or documents), that role is inherited downward through the hierarchy unless it is specifically blocked. This makes it easier to manage access within a document library or an area of the portal.

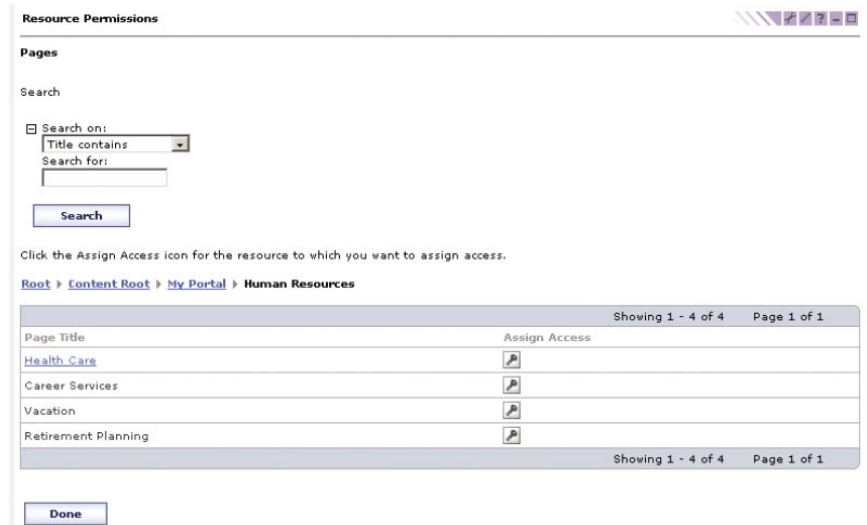


Figure 22. Managing access rights in the portal

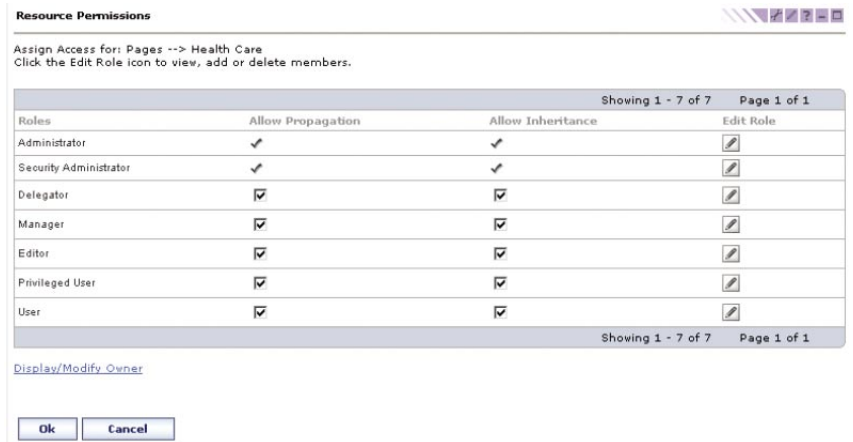


Figure 23. Managing access rights in the portal

Granting *view access* to a page or place means that other users can see pages and places when they log in. Granting *view access to a portlet* means that users can add it to their pages when they customize their portal experiences. Granting *edit access* means that a user can set the portlet settings or change the contents of a page. *Manager access* means that a user can perform view and edit operations, and can delete the portlet or page.

Granting view access to administration portlets is an effective way to delegate certain administrative tasks that are not protected by portal access control to other portal users. Those users can simply add the administration portlets to their personal pages and then perform whatever task the portlet is designed to do. As a result, users do not have to be granted all the administrative privileges or be added to the portal administrator's group. Their administrative abilities are limited to only those tasks covered by the authorized portlets.

Personalizing the portal

Optimizing each user's portal experience is one of the primary goals of WebSphere Portal. To this end, WebSphere Portal provides user and administrative interfaces to customize the content of portal pages, as well as the look and layout of the pages (see Figure 24). With these tools, users can personalize their pages by selecting portlets and changing the settings of each one to suit their specific roles or work habits. Users can also change the page layout and the color scheme (if the administrator enables this option).

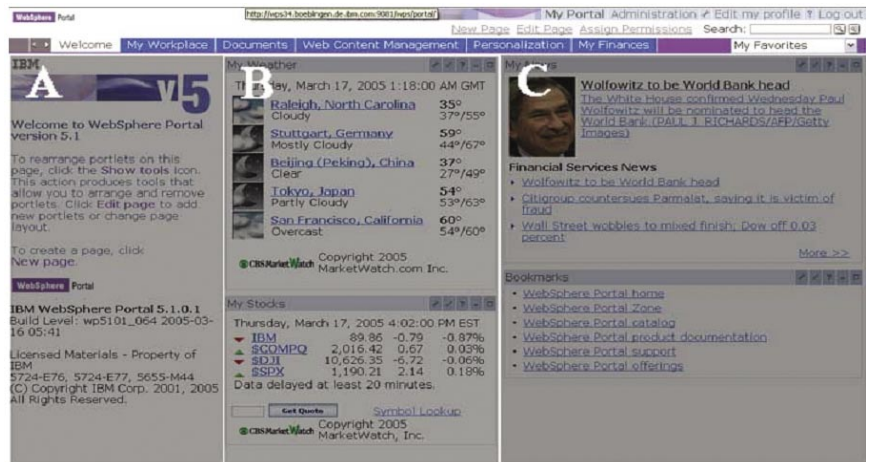


Figure 24. Portal page elements

Users can have one or more personalized pages, navigating to each one from the home page. Pages are arranged in a hierarchical manner with virtually any level of depth. You can assign each page its own color themes, skins and page layouts. *Themes* define the fonts, colors, spacing and other visual elements, and consist of cascading style sheets, JSP files and images. *Skins* are decorations and controls placed around portlets, such as title bars, borders and shadows. At each level of the page hierarchy, the lower pages can inherit the themes and skins from the page above them or can override one or both. Because each section can have a completely different look, you can use sections of the site to create the appearance of different sites running on one portal server.

Each personalized page can have a different set of portlets. These portlets can be selected by end users or by administrators, depending on their access rights for the page. Administrators can specify that certain portlets are required, so that end users cannot remove or rearrange them. Pages can also be rearranged to get a different navigation order for each user or group of users, when permitted by the administrator.

For end users, the portal provides a quick-customizer interface to add and rearrange portlets (see Figure 25). This interface is accessed by clicking the edit layout link at the top of the page you want to customize. Then it is just a matter of clicking the *add content* button to add new portlets.

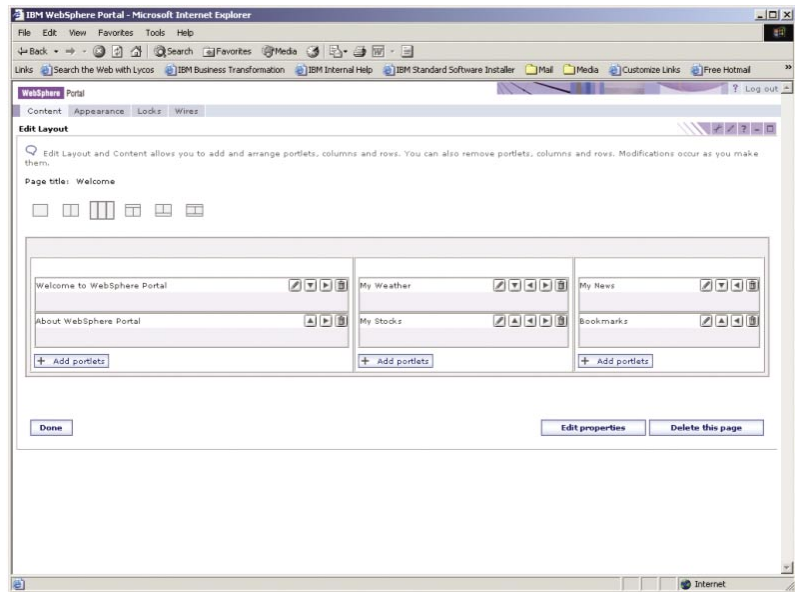


Figure 25. Quick-customizer interface

If you want to rearrange portlets on the page or delete portlets, you can do that inline by turning the page editing tools on and off and selecting the direction icons or delete icon associated with each portlet (see Figure 26).

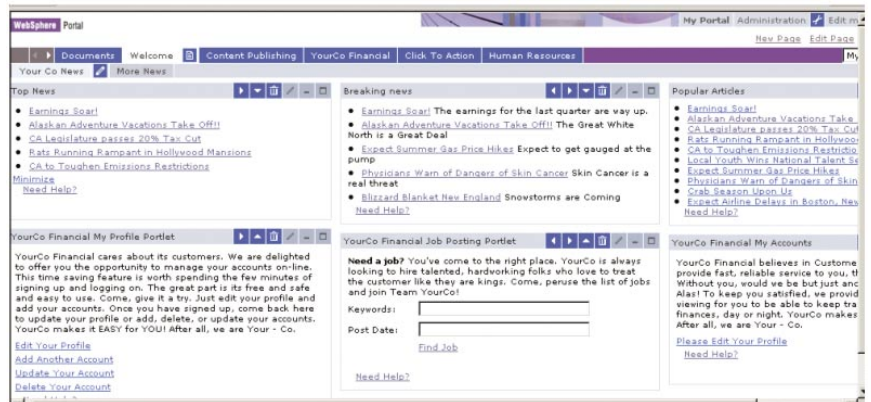


Figure 26. Portlet display customization

A more-advanced page customizer is available for administrators, where the basic structure of a page is defined. Page layouts are fully dynamic. You have the ability to create any arrangement of rows and columns, or to split them to create new rows and columns. Columns can have a fixed size, specified in pixels or in a percentage of the overall page size, which you can adjust by using the width controls. To place portlets on the page, you can click the add content button and search for a portlet you want to add to the page, and use page layout controls to customize placement of portlets to pages.

Personalization

WebSphere Portal offerings include the personalization engine, which enables you to target content to specific users in support of your portal's business goals. The WebSphere Portal personalization component provides facilities that allow subject-matter experts to select content suited to the unique needs and interests of each site visitor. These Web-based tools help companies quickly and easily leverage content created by line-of-business (LOB) users and subject-matter experts.

A personalization solution involves three basic components:

- *A user profile, which provides information about users of the site, including user attributes*
- *A content model, which defines attributes about the content, such as product descriptions, articles and other information*
- *Matching technology, which includes engines that match users to the right content and also provides filtering, rules or recommendation engines or combinations of all three*

The WebSphere Portal personalization engine and WebSphere Portal share a common user profile, a common content model and the JSR 170 content repository. This model is based on the WebSphere resource framework interfaces classes. As a result, personalization rules can easily be added to portlets to select portal content from IBM® Workplace Web Content Management™ and WebSphere Portal Document Manager and target it to the portal's registered users. A set of wizards is included with Rational Application Developer to access Structured Query Language (SQL) or LDAP data.

The basic steps associated with personalization involve classifying site visitors into segments, and then targeting relevant content to each segment. Business experts create the rules for classifying users and selecting content, using Web-based tools (see Figure 27).

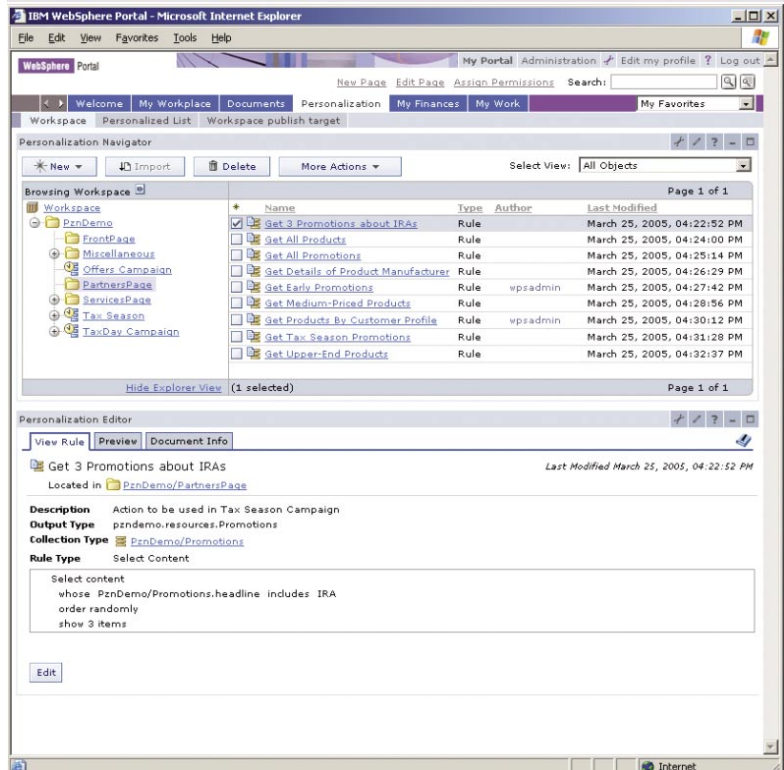


Figure 27. Creating business rules for personalization

The WebSphere Portal personalization recommendation engine provides collaborative filtering capabilities. Collaborative filtering uses statistical techniques to identify groups of users with similar interests or behaviors. Inferences can be made about what a particular user might be interested in, based on the interests of other members of the group.

The WebSphere Portal personalization engine also includes campaign-management tools. Campaigns are sets of business rules that work together to accomplish a business objective (see Figure 28). For example, a human resources (HR) manager might want to run a campaign to encourage employees to enroll in a stock purchase plan. The HR manager would define a set of rules as shown to accomplish this business objective. Campaigns have start and stop dates and times and can be e-mail- and Web-page-based. Several campaigns can run simultaneously and can be prioritized.

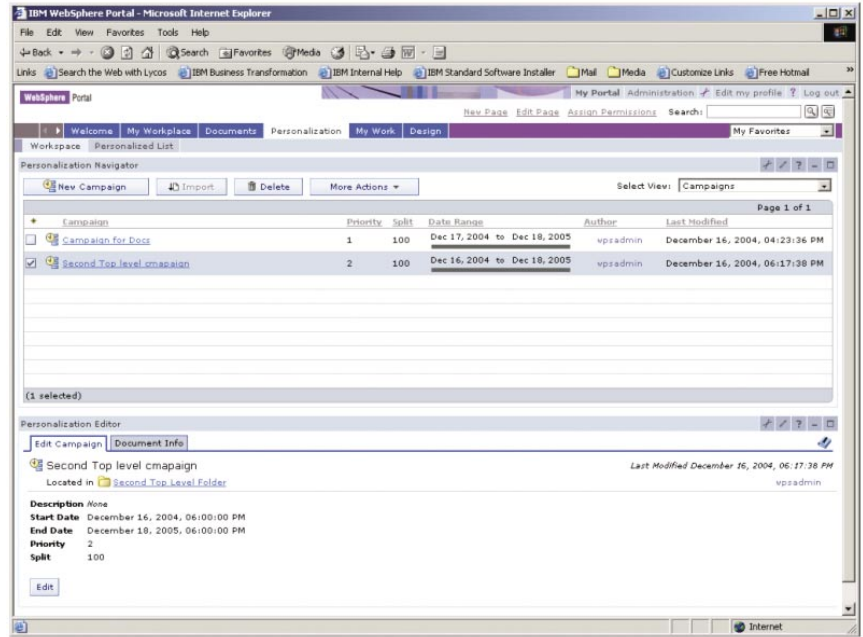


Figure 28. Setting up campaign parameters

Implicit profiling services can collect real-time information about site visitor actions and then construct personalization business rules using this data. Implicit profiling tracks the areas of a site where a user is most active to determine the user's interests. For example, if a user clicks articles about football, it is possible to determine that he is interested in sports, even if he did not explicitly indicate this interest when registering at the portal. The personalization rules and campaigns are stored in the WebSphere Portal Java Content Repository (JCR) store, and you can write rules to personalize the WebSphere Portal Document Manager or IBM® Workplace Web Content Management™ content right out of the box (in other words, without developing your own resource collections).

To analyze the effectiveness of the site and its personalization strategies, the server provides a logging framework that can be enabled using WebSphere Portal administrative services. These logs can be analyzed by Tivoli Web Site Analyzer, included with IBM WebSphere Portal Extend. Tivoli Web Site Analyzer can then create reports for the portal's business owner. This helps your company measure the effectiveness of the business rules and campaigns in achieving their objectives. Personalization rules and campaigns are stored in the Java content repository, which enable you to, out of the box, write rules to personalize JCR content.

Cascading authorization for portal customizations

Many companies need to build portals that accommodate the needs of both a central organization as well as regional and local communities of users. For example, you might want to have top-level design decisions made centrally, such as the visual elements and basic page layouts that reflect corporate image and brand identity. Divisions or other organizational units might need the ability to add their own pages and content, or perhaps even subdivide areas of the common pages.

WebSphere Portal supports the needs of companies that want to cascade portal definitions across the organization using derived pages. Top-level administrators define base pages, and then permit subordinate administrators to further refine the page layouts and content to meet their individual needs. The refinement process can continue for any number of levels, until it finally reaches users (see Figure 29). If permitted, end users can customize their own pages by selecting and placing portlets and by changing portlet settings. To compute the page that a user finally sees, the portal server merges the page fragments defined by each successive refinement.



Figure 29. WebSphere Portal supports delegated administration of portlets and pages.

Administrators at different levels of your organization can lock the layout or content of any area of the page. For example, locking the placement of portlets means that users cannot move them or remove them from their pages. As a result, a higher-level administrator can set up the basic structure of the page, and fix certain portions or leave others open to modification by other administrators or by end users. Lower-level administrators cannot override any restrictions imposed by higher-level administrators. This capability means that your company can achieve both horizontal administration (across the company, business segments, locations and users) and vertical administration, where administrators provide content dedicated for a specific page. Using page locks with WebSphere Portal access-control features, administrators can determine which content is:

- **Mandatory.** *Users cannot delete the content from their pages.*
- **Recommended.** *The content is placed on the page initially, but users can delete it.*
- **Optional.** *The content is not placed on the page, but is available for users to add it.*
- **Forbidden.** *Users cannot see this content at all.*

Skins and themes

The portal server uses a system of JSP templates, cascading style sheets (CSSs) and images to define the look of the portal pages. You can modify these to control any of the visual aspects of the portal, perhaps to add company-specific brand elements, or to achieve a different color scheme and visual style. The system for defining color themes and portal skins has been enhanced to support multiple skins per theme, other branding elements, navigation styles and dynamic, browser-independent cascading style sheets (see Figure 30).

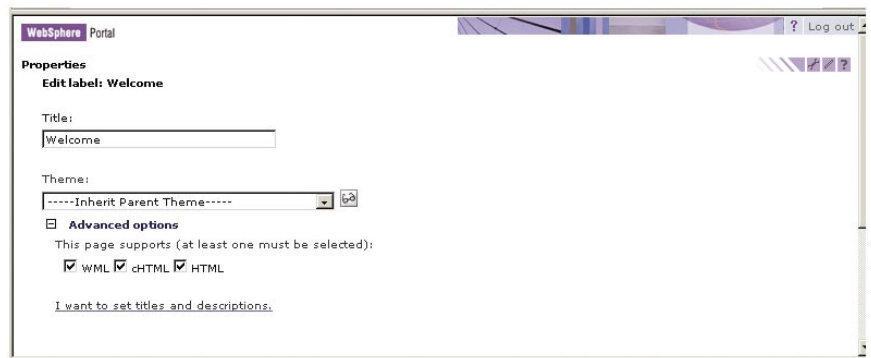


Figure 30. Setting themes and skins

Skins and themes can now be applied to each page rather than being restricted to having only one for the overall portal. Different skins can be applied individually to portlets, so that the portal look can be fine-tuned to meet any needs (see Figure 31 and Figure 32).

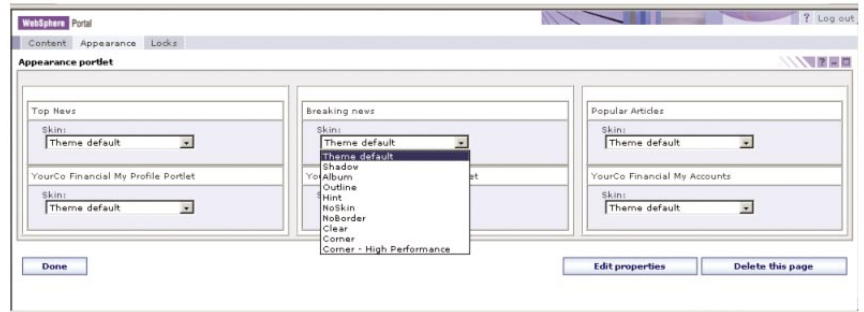


Figure 31. Managing themes and skins

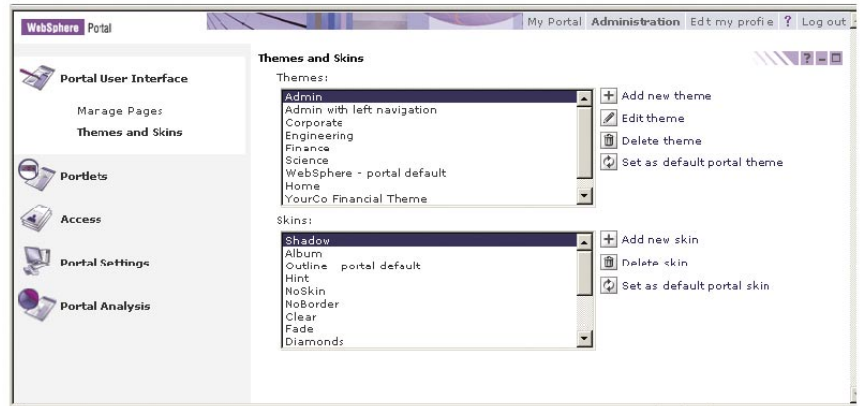


Figure 32. Managing themes and skins

All of the visual elements of the portal, including the masthead, the navigation areas, graphics, portlet title areas and style sheets can be changed to give the portal a custom look, as shown in the example in Figure 33. Standard file formats, such as JPEG, GIF, CSS and JSP files are used to define the look and layout of the portal. If you examine the structure of the portal server's installation folder, you notice folders named *skins* and *themes*, with folders named *html*, *wml* and *chtml* within them. These folders contain most of the files used to define the basic structure of the portal home page, its color schemes and portlet decorations. You can make copies of these folders and modify the contents to achieve the visual image that your company needs. The theme administration portlet registers the new files.



Figure 33. Customizing branding and appearance of your portal using the themes and skins included in WebSphere Portal

Navigation

WebSphere Portal navigation capabilities have been improved significantly. Now, instead of having a simple page list for each place, WebSphere Portal makes it possible to create very complex navigation trees, with both page links and links to external URLs shown in the navigation area (see Figure 34 and Figure 35). You can create labels that logically group certain navigation elements, with more links indented beneath these labels.



Figure 34. Page-navigation administration

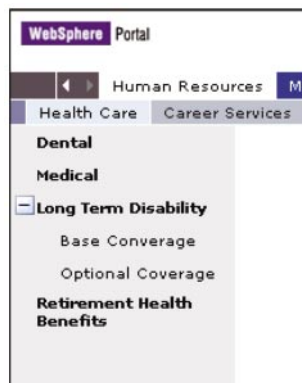


Figure 35. Page-navigation view

WebSphere Portal release builder

WebSphere Portal, Version 5.1 includes a configuration-management process, called a *release builder*, that enables you to efficiently track and apply changes to new applications from staging to production portal servers. Using the WebSphere Portal release builder, you can export configuration entity modifications to an XML file, which tracks entities that were changed, added or removed, and then applies the differences to your production environment. The result improves the ability to keep staging and production systems synchronized as updates are applied to the production portal server.

Virtual portals

WebSphere Portal facilitates the efficient usage of hardware resources by enabling your organization to create multiple portals that share the same WebSphere Portal software and hardware implementation (see Figure 36). You can create and manage varying business requirements and constituent audiences by creating and managing more virtual portals without running more WebSphere Portal installations. For example, as a service provider, you can provide different portal services for multiple customers and their users.

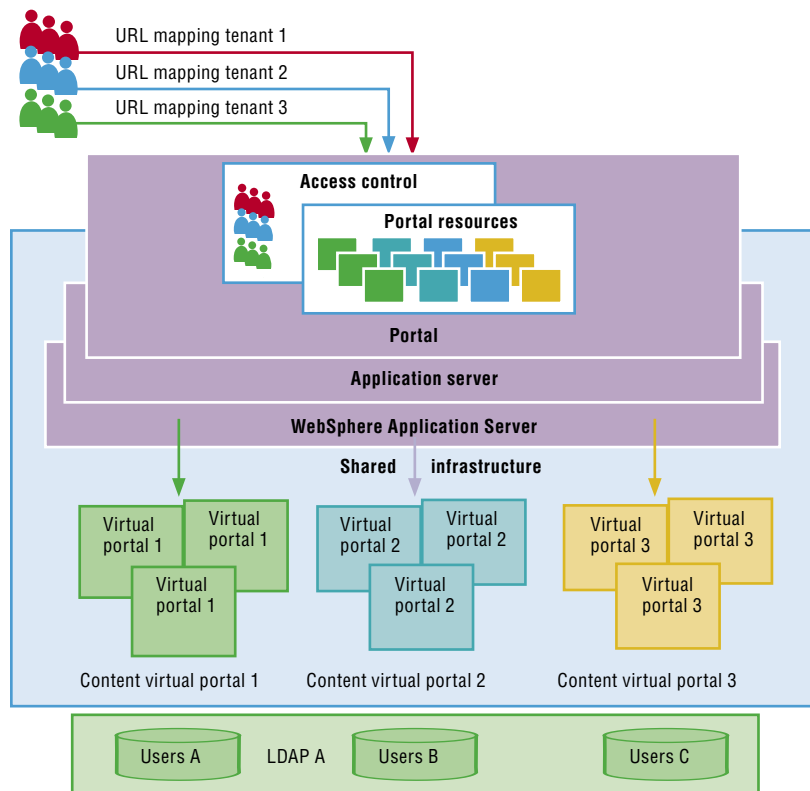


Figure 36. Multisite implementation using virtual portals

WebSphere Portal, Version 5.1 provides new specific administration portlets that enable the setup of new virtual portals and management of existing ones. A virtual portal is accessed through a specific URL mapping, associated with that virtual portal. Resources for each virtual portal are separated using WebSphere Portal access control, including portal configuration and page hierarchy, user populations, administration and anonymous pages. Delegation of independent WebSphere Portal access-control management within virtual portals is also supported. To learn more about virtual portals, visit the WebSphere Portal, Version 5.1 information center at ibm.com/developerworks/websphere/zones/portal/proddoc.html.

Universal access

This white paper previously discussed WebSphere Portal support for international use. The entire system of page templates, themes, skins and portlet rendering is fully enabled for internationalization (including double-byte and bidirectional languages) and for accessibility to people with disabilities (see Figure 37). The portal server generates markup that complies with the American Disability Act (ADA) as defined in “Section 508 Web Accessibility Standards” and meets the guidelines of the Worldwide Web Consortium (W3C) Web Accessibility Initiative.

For globally accessible portals, the portal server searches for and selects the correct JSP based on the target browser, and the target browser’s settings for language and country. To enable a portlet for language selection, its JSP files are packaged using the following directory structure:

```
WAR/markup_type/language/country/variant/jspname.jsp
```

WebSphere Portal searches for the JSP for a portal from the most specific to the least specific, as in the following example:

1. */html/en/US/IE/portlet.JSP.jsp*
2. */html/en/US/portlet.JSP.jsp*
3. */html/en/portlet.JSP.jsp*
4. */html/portlet.JSP.jsp*
5. */en/US/IE/portlet.JSP.jsp*
6. */en/US/portlet.JSP.jsp*
7. */en/portlet.JSP.jsp*
8. */portlet.JSP.jsp*

Another good technique for handling translation issues is to use the portal server's JSP tag library. It includes a tag for identifying and separating translatable text, so you don't have to create and maintain separate JSP files for each language. Using this technique, you could create one JSP view, tagged like this:

```
<H2><wps:text key="heading.hello" bundle="mystrings"/></H2>
```

When this page is processed, the strings are substituted using standard Java resource bundles.

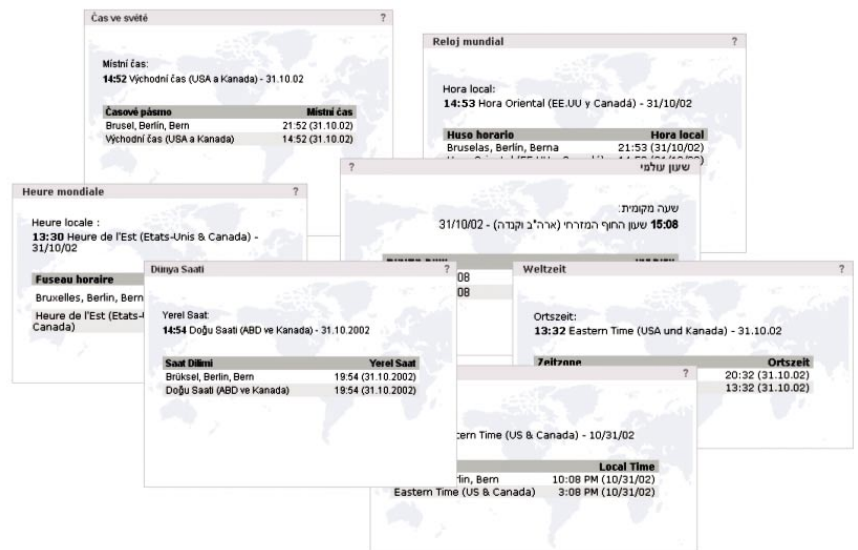


Figure 37. Portlets in many languages

For portals that need to support many languages, or for portals that include rapidly changing portal content, IBM WebSphere Translation Server, included with WebSphere Portal, offers automatic translation technology. This technology is useful because it supports either real-time or offline translation of human languages. WebSphere Translation Server provides quick, inexpensive, convenient text translations that communicate at a minimum the text's essence whenever a professional translation is not feasible because of availability, time or cost. Dictionaries can be tuned so that idiomatic expressions and specialized terminology are interpreted correctly. WebSphere Translation Server also supports bidirectional translation of content between English and French, Italian, German, Spanish, Japanese and Chinese (simplified and traditional). Unidirectional translation of content from English to Korean and Brazilian Portuguese is also supported.

Using portlets to administer your portal

Administration of the portal is done through the portal itself, either in a centralized or delegated fashion. Administrators can deliver a new service to users simply by adding new portlets to the pages of the portal. Because these are portlets, just like bookmarks or reminders or news or any other portlets, administrators can control access to them, place them on portal pages and perform any of the usual steps.

Administrative portlets are provided to add portlets to the portal's registry, manage users, groups and access-control lists, integrate Web content with clipping technology, configure Web services-enabled applications, set portal-wide settings, manage logs and other common tasks. The next section describes some of the administration portlets and what they do for you. Because some of these portlets have already been described in the previous section, this section focuses on those that have not been covered already.

Global settings portlet

In the global settings portlet, administrators can change portlet settings such as the default language, the cache timeout values and so on (see Figure 38). Settings are also available that control how new user sessions are handled, and what to do when a user tries to access a portlet without authorization. Unauthorized access can be ignored (in other words, the portlet is not displayed), or the portlet can be replaced by an informative message so that the user can take the necessary actions to correct the situation. Returning users might want to pick up where they previously left off, so there is a setting to retain the state of the last visit and return to that page the next time.

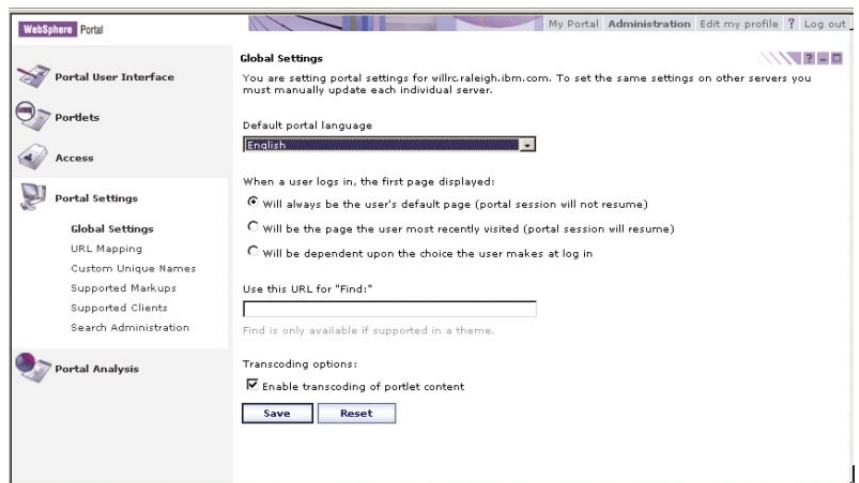


Figure 38. Global portal settings

Web clipping portlet

One of the most important portlets is the Web clipping portlet, which is used to display sections of existing Web pages (see Figure 39). You can visually select portions of the page or clip all the text between specific tags. This method enables you to precisely control what markup is extracted and quickly enable access to external Web-site content sources from within WebSphere Portal. The portlet can optionally rewrite the links inside the clipped page, which is useful for displaying existing pages without leaving the portal's navigation structure. Each time you clip a Web page, a new portlet is created in the portal's registry. Whenever the new portlet is displayed, it retrieves the current version of the Web page and extracts the clipped portion to display.



Figure 39. Clipping Web pages

Some sites that you clip might require authentication. The Web clipping portlet provides options for no security, basic authentication, form-based or Microsoft Windows NT® local area network (LAN) manager (NTLM) authentication. The credentials can be provided by the user or filled in by the administrator.

Web services

A *Web service* is an interface that describes a collection of network-accessible operations. WebSphere Portal provides support for the OASIS WSRP standard. The OASIS WSRP standard simplifies the integration of remote applications and content into portals. With WSRP, you can select from a rich choice of remote content and applications (presentation interfaces and their data) and integrate them into your portal with just a few mouse clicks and no programming effort. You can also configure WSRP producers support for specified WSRP-enabled local portlet applications as presentation-oriented WSRP services and make them available to remote portal server consumers who want to use these services.

The WSRP interface is described using a standard XML description language called *Web Service Description Language (WSDL)*, so that the service can be invoked without prior knowledge of the platform, language or implementation design of the Web service. Web services are located using the standard Universal Description and Discovery Interface (UDDI), which can be a private or public registry of Web services.

The Web services administration portlet enables administrators to configure Web services producer and consumer support for specified WSRP-enabled applications within the portal (see Figure 40).

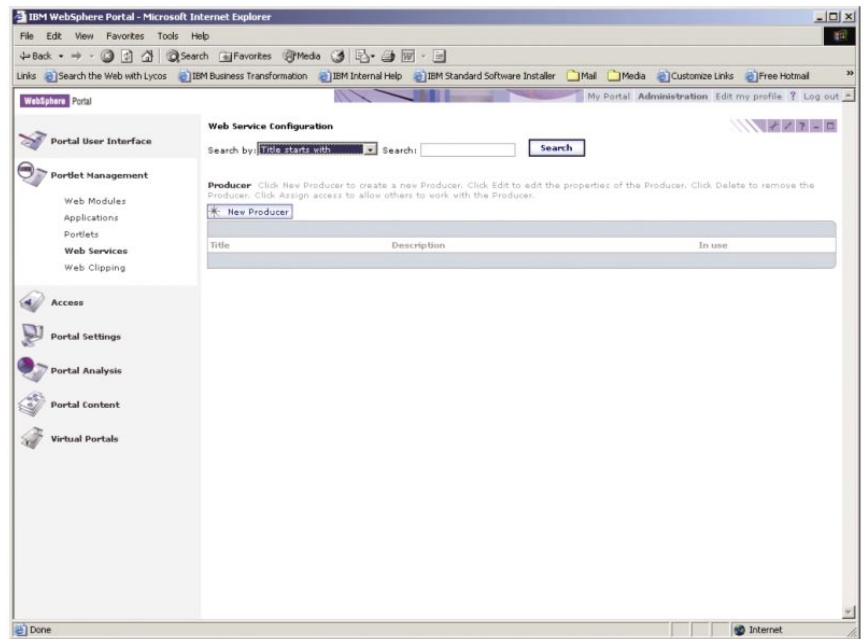


Figure 40. Configuring WSRP producer services

Support for WSRP also enables WebSphere Portal to integrate external applications through remote portal servers, and to promote local portal portlet applications to remote portals. A new portlet can be automatically activated, but with no special permissions. Access control for the new portlet is inherited from the default portlet settings, but can be set explicitly by the administrator (see Figure 41 and Figure 42).

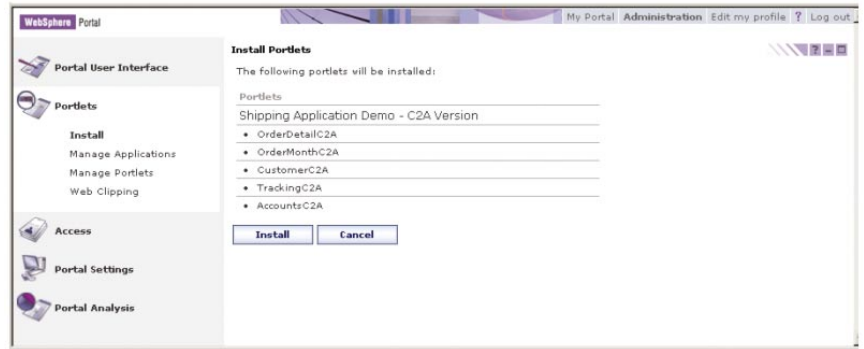


Figure 41. Installing new portlets

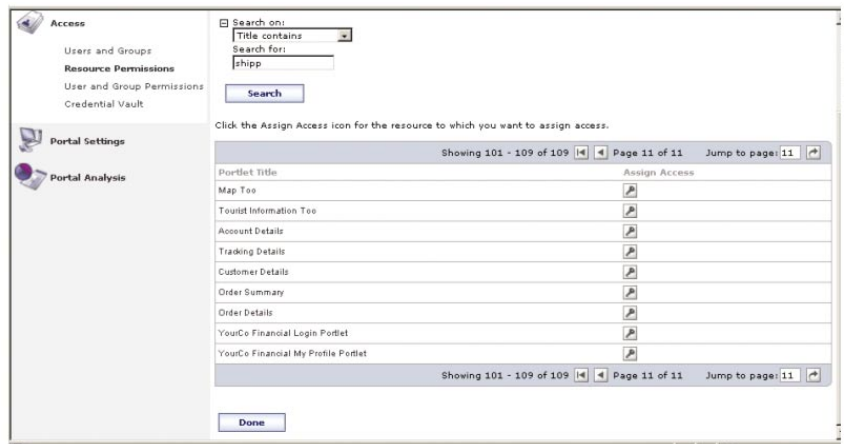


Figure 42. Administrative access controls

After the portlet is installed, you can copy it, set its configuration parameters, activate or deactivate it, or uninstall it (see Figure 43).



Figure 43. Managing portlets

User and group portlets

You can use the portlets provided to manage user and group information without leaving the portal (see Figure 44). You can also manage a user's group memberships. These portlets provide search capabilities and pagination that enable the administrator's interface to scale and manage a large number of users and groups.

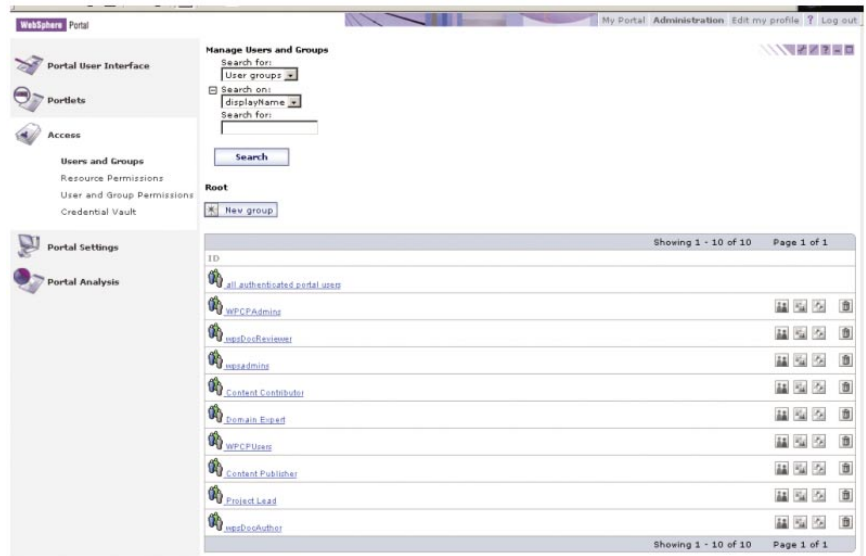


Figure 44. Managing user and group memberships

The portal server uses group-membership information to determine what pages, portlets and documents a user is authorized to view and edit. Users can be members of one or more groups, and groups can contain other groups. Users are allowed access to portal resources when access is granted to any group to which the user belongs. Access rights can also be granted to specific individuals, but most companies find that it is easier to manage the access rights of groups instead.

Portal analysis and logging portlets

You can control the tracing and logging activity and monitor user activity through the portal analysis administrative options and portlets that enable tracing and track frequent-user information. You can also control tracing and logging activity by modifying the configuration properties files of the logging subsystem. The portal server also records user activity in logs that can be processed by Tivoli Web Site Analyzer (see Table 2 and Figure 45). Overall usage statistics such as logins and logouts are tracked, along with portlet and page-usage statistics.

Model	Advantage
Project summary	Provides presentation, user management, security and other services for constructing a portal.
User-agent ranking	Displays a ranking of user agents employed by visitors to your site.
Browser ranking	Displays a ranking of browsers used by visitors to your site.
Platform ranking	Displays summary statistics of overall platform activity related to new users, pages, portlets and modifications from portal log analysis.
User ranking	Displays a ranking of users to your site by frequency of visits.
Page ranking	Displays a ranking of the pages viewed or edited by visitors to your site.
Portlet ranking	Displays a ranking of the portlets viewed by visitors to your site.
Commands ranking	Displays a ranking of the commands run by visitors to your site, such as editing pages, creating new pages, and creating or deleting users and groups.
Logins	Displays the number of logins and logouts into the portal.
Resource ranking	Displays a ranking of the resources viewed by visitors to your site.

Table 2. Portal usage reports from Tivoli Web Site Analyzer

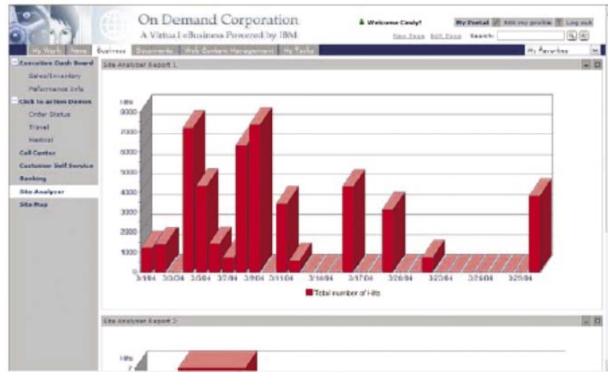


Figure 45. A portal usage report from Tivoli Web Site Analyzer

IBM Common PIM portlets

WebSphere Portal, Version 5.1 introduces IBM Common PIM portlets, designed to provide a common portlet interface to interactively work across a number of popular mail and personal information management (PIM) applications, including Lotus Notes and Domino, Microsoft Exchange, IMAP and POP3 mail applications. Common PIM portlets integrate with WebSphere Portal applications and services, including the WebSphere Portal rich-text editor, to create and edit mail messages and documents (see Figure 46). The portlets include the ability to view received attachments using the portal attachment viewers and the ability to store and retrieve documents from the WebSphere Portal Document Manager. Presence-awareness capabilities are also available, to provide access to instant messaging and chat services.

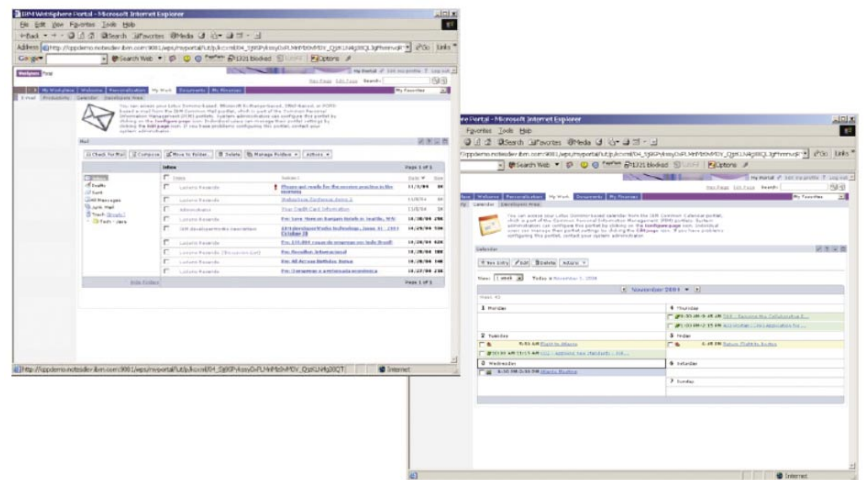


Figure 46. The IBM Common PIM portlets for mail and calendar

Application-integration portlets

By definition, a portal provides access to content, data and services located throughout the enterprise. These include not only predefined connectors and portlets, but also other connectors and portlets that can be created by various tools. ERP and CRM systems are excellent candidates for these portlets because efficient, personalized access to these functions provides measurable return on your portal investment.

WebSphere Portal includes portlets and portlet builders that help you access a variety of ERP and CRM systems. IBM WebSphere Portal Application Integrator is a nonprogrammatic, wizard-driven portlet-building tool, provided with WebSphere Portal, that offers a fast, efficient way for power business users to build new portlets that access various kinds of enterprise application systems, including portlet builders for SAP, Siebel, Oracle and Domino applications, and Java Database Connectivity (JDBC) sources. It also works for relational databases, such as DB2, Oracle and others. WebSphere Portal Application Integrator portlet builders work by querying the back-end system for metadata that describes the business object of the specific enterprise application that the new portlet is meant to work against.

After the business object is selected, the person building the portlet selects which fields and operations to enable in the portlet. These operations include searching, updating, deleting and creating new records, as shown in Figure 47.

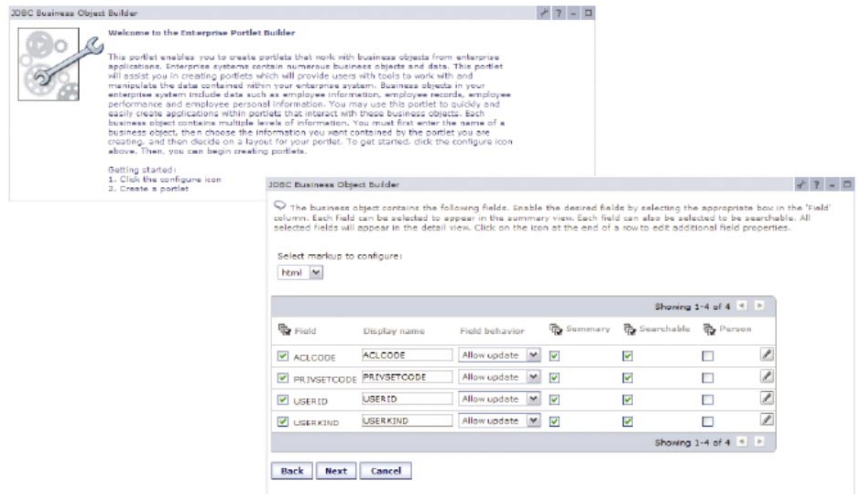


Figure 47. Creating an enterprise portlet with WebSphere Portal Application Integrator using the JDBC builder

Besides the ability to select application fields to present in the portlet interface while using the WebSphere Portal Portlet builders, users have the option to select the person field. Users can then match that field to the corresponding back-end application field to be enabled for presence awareness, instant messaging and chat services within the enterprise portlet – without additional programming.

WebSphere Portal Application Integrator portlet builders also provide advanced field options to enable selected portlet fields for data exchange with other portlets and applications using click-to-action, and the ability to access user-profile data to prepopulate portlet fields with user data (see Figure 48).

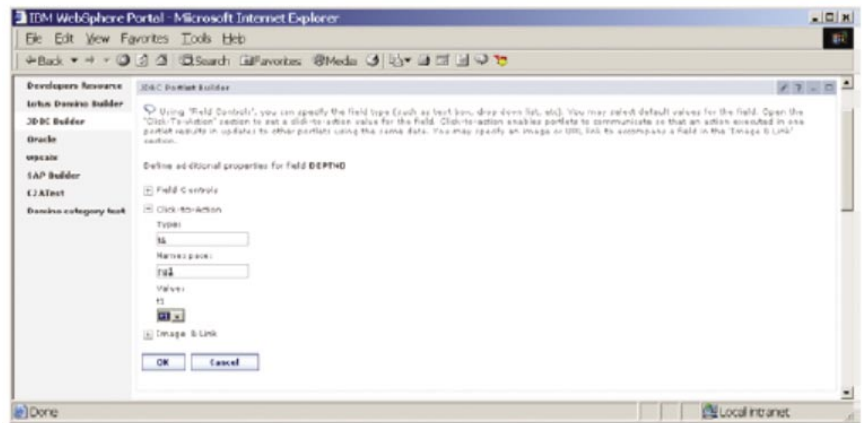


Figure 48. Enabling click-to-action field properties using the WebSphere Portal Application Integrator JDBC builder

WebSphere Portal Application Integrator portlet builders support the WebSphere Portal credential vault service to store and retrieve users' credentials and pass them on to the enterprise application and business object to manage single signon for the portlet user with the respective enterprise application. After the selections are made, the portal server stores the configuration information needed to run the portlet. No new Java code or service needs to be deployed, just the configuration information for each portlet.

Using this approach, anyone who understands the usage pattern of the enterprise application system can build a portlet in a very short time, and optionally include presence awareness and data exchange between portlets and enterprise applications using click-to-action, as is shown in Figure 49.

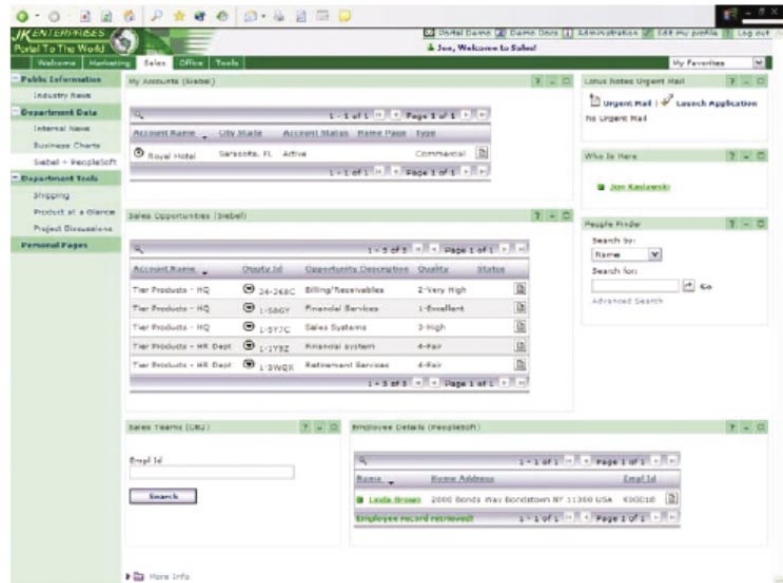


Figure 49. WebSphere Portal Application Integrator technology-generated portlets for DB2 and Siebel applications, enabled with presence awareness and data exchange using click-to-action

Collaboration portlets

Corporate portals connect people to the applications, content and resources they need. Portals also connect people to each other, through community pages, shared bookmarks, contact lists and personal productivity portlets. Collaboration is really about people working efficiently and effectively in teams to solve business problems. WebSphere Portal includes out-of-the-box, preconfigured pages, portlets and services designed to support the team and its activities, with messaging, document libraries, user profiles, inboxes and calendars, online directory services, team workplaces and electronic meetings. Users can access these collaborative services in the context of what they are currently doing, instead of having to leave the portal to open another application.

A portal user can see directly from the portlet if other users are online and select from a menu of options to interact with another user. For example, while working in the portal, users can easily see who else is online and then send an instant message, send an e-mail or add a person to their contact lists. Collaborative portlets have advanced built-in features that allow portal users to take actions on documents or user names that appear in a portlet.

Lotus Collaboration Center is a set of preconfigured pages and portlets to provide out-of-the-box integration with Lotus Notes and Domino and collaborative products. The Notes and Domino portlet offers access across mail, calendar, to-do list, other Notes application views, team room and discussion. The IBM Lotus Domino Web Access (IBM Lotus Domino iNotes™) portlet offers mail, calendar, to-do list, contacts and notebook journals. Portlets are also provided for Lotus Domino Document Manager, Lotus Web Conferencing, Lotus Team Workplace, Lotus Instant Messaging and the people-finder directory-search capability.

Using these components, users have immediate access to key team-discussion, document-management, team-room and mail applications, and a searchable directory of people that is integrated with their workplaces and their e-meetings within the collaborative portal. Users can find people in the directory, see their online status and interact with them using instant messaging and other actions provided by people links. Besides search features, the people-finder portlet provides views of each person's directory record and his or her place in the organizational context. Figure 50 shows an example of a Lotus Domino Web Access portlet. Figure 51 show examples of a Lotus Collaboration Center portlet configuration.

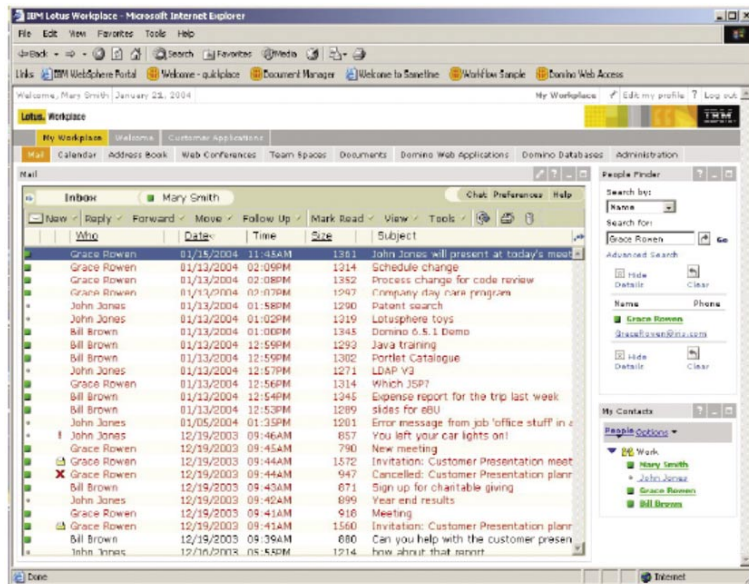


Figure 50. Example of the Lotus Domino Web Access portlet

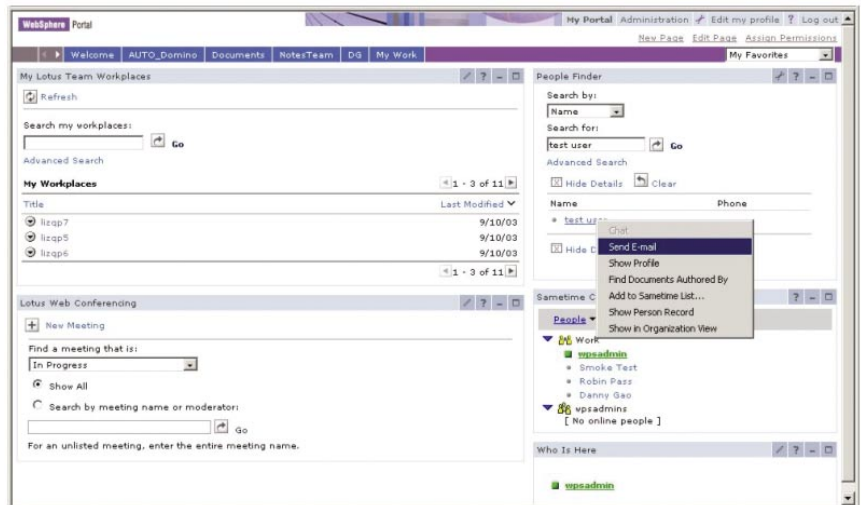


Figure 51. An example of WebSphere Collaboration Center pages and portlets

Portlet	Advantage
Domino Web Access (iNotes)	Provides access to an IBM Lotus Domino Server using the Domino Web Access interface. This portlet can display Domino Server mail, calendar, to-do list, contact and notebook functions.
Lotus Notes and Domino	Provides access to Notes mail, calendar, to-do list, additional Notes database views (optionally across Notes databases), team-room and discussion databases.
Domino Document Manager	Displays a Domino Document Manager library inside the portlet.
Lotus Team Workplace (QuickPlace)	Helps users find, locate, request new team workplaces and work in team workplaces.
Lotus Instant Messaging contact list	Displays a list of people who are online and available for chat, either in the current page or in a named virtual page.
People Finder	Locates a person by name or profile characteristics, and provides quick-search and advanced-search features.
Web conferencing	Lets users schedule, find and attend e-meetings, and shows e-meeting details.
Who is here	Displays list of names for chat, and enables users to modify groups for display and shows a status message.

Table 3. Lotus Collaboration Center portlets

IBM WebSphere Portal Extend integrates tightly with Lotus Domino and IBM® Workplace™ world-class collaboration products including Lotus Team Workplace and Lotus Instant Messaging and Web Conferencing, by providing portlets and services to access these products seamlessly from the portal. Lotus Team Workplace provides workspaces for sharing and organizing ideas, content and tasks. It includes tools to manage team projects and schedules, organize discussion threads and share documents. The IBM Lotus MyTeam Workplace user interface is shown inside a portlet. The integration between the portal and the team workplace includes single signon, so that users who access the portal can access team workplaces seamlessly, without having to log in more than once.

Lotus Instant Messaging provides instant messaging, shared white boards and application sharing for e-meetings. Lotus Instant Messaging functionality integrates with the portal to enable access to chat sessions and buddy lists, as well as people and place awareness (the ability to tell who the members are and to find out whether they are online, offline or not available).

Collaborative services

IBM Lotus collaborative components are building blocks (APIs and JSP tag libraries) that can be used to integrate the functionality of Lotus Domino, Lotus Instant Messaging and Lotus Team Workplace into the portal. Developers can leverage the features of these products by using components to add user interface extensions to their portlets and portal pages. These collaborative services hide the configuration details of the Lotus products installed within your enterprise. Instead of working with more-complex product APIs, Lotus collaborative components and services APIs give you an easier method to integrate core collaborative features into any portal or portlet.

The benefit of using collaborative components is that they provide standardized access to Lotus applications, with easier APIs that are optimized for the portal. The security context is handled automatically, and upgrades of the back-end systems are transparent to the portlets. The collaborative components work across IBM Lotus Domino 4.67 and later. Table 4 summarizes the collaborative services and collaborative components available through WebSphere Portal.

Model	Advantage
People services	Provides access to an IBM Lotus Domino Server using the Domino Web Access interface. This portlet can display Domino Server mail, calendar, to-do list, contact and notebook functions.
Lotus Notes and Domino	Provide online awareness capabilities so that portlets can determine who is online and what their status is. Also incorporate menus to support chat, e-mail and search options.
Domino data services	Provides access to Domino data, including views, documents and directory-related information. Enables you to locate Domino servers, create and edit documents and work with views.
Discovery services	Provide methods to find people, documents, knowledge maps and profiles.
Workspace services	Provide methods for creating new Lotus Team Workplace workspaces.
Instant messaging services	Provide standardized access to the Lotus Instant Messaging server configured to work with WebSphere Portal.

Table 4. Lotus collaborative services APIs component-services available through WebSphere Portal

People services provide awareness functions that are available through a custom JSP tag. Adding this custom tag creates an online status indicator and a pop-up menu of action choices that access various collaborative services. For example, you could start with a portlet that produces markup like this:

```
<%@page language="java" import="com.lotus.cs.*, java.util.*" %>
<%@taglib uri="*/WEB-INF/tld/people.tld" prefix="peopleservice" %>
<%@taglib uri="*/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init/>
<h1>People Awareness Example</h1>
<%
CSEnvironment environ = CSEnvironment.getEnvironment(request);
CSCredentials cred = environ.getCredentials(request);
%>
<H2>
<peopleservice:person>
<%=cred.getUserLoginName()%>
</peopleservice:person>
</H2>
```

The highlighted lines are the special tags needed to make the online presence indicator work. The portal user might see something similar to Figure 52.



Figure 52. People awareness

A good way to apply this idea is to add the people-awareness tags to a portlet that accesses an enterprise application, such as a Siebel function. The exact menu items depend on details about the specific place, the permission level of the current user, and what collaborative products and services are installed in the portal environment. All of the collaborative services except for the people-services tags can be used in both desktop and mobile portlets.

Business-process integration and task management

In an on demand business environment, where efficient interaction among customers, employees and suppliers is key to success, you must enable these parties to seamlessly participate and collaborate in business processes. The role-based, user-facing tasks within these business processes – such as completing forms, forwarding for other approvals or transaction completion – must be presented to the right people at the right time in a way that allows them to react quickly and that enables processes to run efficiently. Not only is a list of tasks presented, but all relevant information and access to the applications a user needs to process them must also be available.

WebSphere Portal extends the ability to easily integrate with enterprise applications through features that exploit the process engine of the WebSphere Business Integration Server Foundation process choreographer. The process choreographer capability enables you to integrate your existing business processes and applications into the portal by exposing role-based access to all user interfaces required to interact with tasks and business processes. The portal can display visual alerts notifying specific users when they are assigned a new task. These alerts link users to the My Tasks portlet, where they can view, claim and launch tasks on a new, dynamically rendered portal page (see Figure 53).

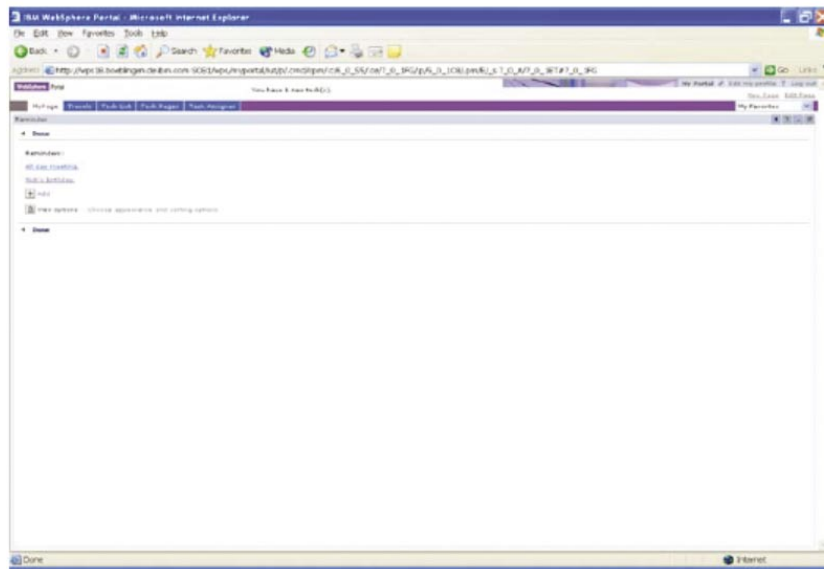


Figure 53. The My Tasks portlet for role-based user notification of new tasks

By clicking a task's title, the user can launch a task page that contains all information and applications to process the given task (see Figure 54).

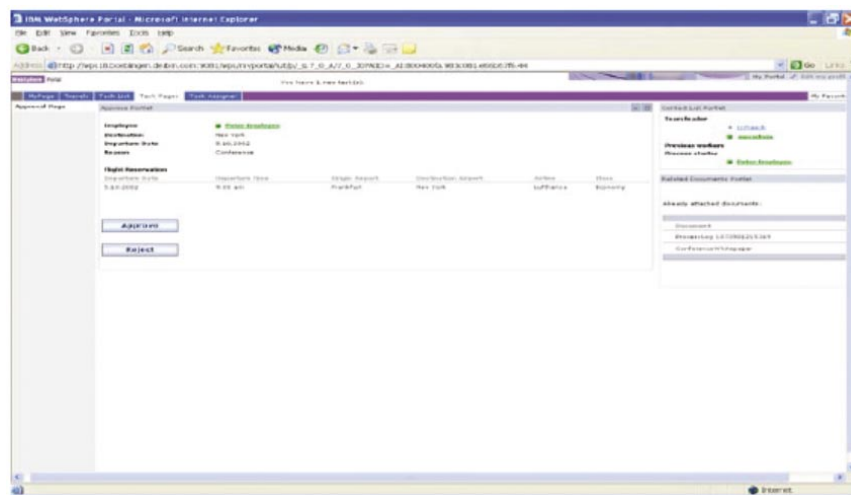


Figure 54. Portal management of business-process tasks

Using the process-choreographer execution engine, WebSphere Portal supports implementation of standards-based Business Process Execution Language (BPEL) business processes, which are modeled and created using IBM WebSphere Studio Application Developer Integration Edition. Business processes typically span several enterprise applications and interactions. This results in the creation and management of composite applications from the portal, providing role-based interfaces that manage user interactions across multiple applications. Transactions across systems are integrated using standards-based, data-transfer connector and communication protocols.

Dynamic UI portals

As more information and services are added to your portal infrastructure, you have an increased need to manage transient applications – required only for a discrete period of time – more efficiently. WebSphere Portal includes support for such pages and portlets through new dynamic user interface (UI) manager capabilities and published APIs.

Dynamic UIs are suited for applications in which users might need to have several instances of a page or portlet open for multitasking. Business-process integration is an example of a dynamic UI configuration. Consider a travel request scenario as an example. If several sales representatives need to travel to a conference, their manager would receive multiple travel requests. With static pages, the manager must complete one request before proceeding to the next one. Using dynamic UIs, the manager can open several requests simultaneously and navigate among these and other pages in the portal.

Dynamic pages and portlets contain many of the same properties as static pages and portlets. For example, the user can navigate between static and dynamic pages, or change the window state of a dynamic portlet. However, dynamic UIs are not stored in the portal database, so they have less impact on server performance.

A *dynamic UI* is a copy of a page or portlet definition at the time the instance is created and is not affected by subsequent changes to the definition. As a result, a dynamic UI is independent of its point of origin. For example, if a portlet launches a dynamic page, the launched page is maintained even if the originating portlet or page is deleted. By default, the dynamic UI acquires the title and description of its page or portlet definition. However, it is possible to programmatically overwrite the title and description when the dynamic UI is launched.

Dynamic portlets include portlets on a dynamic page and portlets that are dynamically added to the page. These dynamic portlets can receive properties from the portlet that launches the dynamic UI.

IBM WebSphere Commerce

Many portals need to include access to commerce functions. These functions include portlets that access a catalog of products and services, or portlets that present a catalog of products and services, and that price, discount, accept orders, process requests for quotes (RFQs) or provide contract-driven views into the catalog to give business customers personalized service. Portals can provide these functions to consumers or to business partners, to facilitate procurement or electronic exchange that links buyers and sellers of goods.

WebSphere Commerce software enables you to build a robust extended enterprise – a network of employees, suppliers and partners collaborating to meet the full spectrum of customer needs by leveraging configurable business processes to speed time to value for customer- and partner-facing projects. The WebSphere Commerce business context engine controls these business processes to present information and functionality to your constituents in a context-sensitive manner. For example, partners have access only to the information for products they are qualified to resell, with parts, inventory information, site language and currency specific to their geographies.

Mobile portals

Next-generation portals must be accessible through more than just traditional desktop browsers. Access through handheld or mobile devices is becoming increasingly important. WebSphere Portal currently supports mobile devices by generating portal pages in three markup languages: HTML for desktop computers and some personal digital assistants (PDAs) and Wireless Markup Language (WML) for Wireless Access Protocol (WAP) devices (typically mobile phones) and compact HTML (cHTML). See Figure 55 for an example of a portal for mobile users.

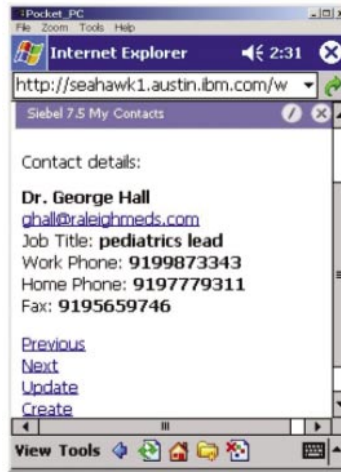


Figure 55. Siebel application portlet settings optimized for mobile users

Users 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, the page is produced by first detecting the type of device that is making the request, and then assembling the portlets, with each portlet rendering its content in the appropriate markup language. When a user customizes the home page for a particular device, the portlet selection list shows only 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 are available for all the supported devices, while others are available only on a single device. The user interface design of each portlet varies from device to device, so that the user's experience can be fine-tuned. Thus, the user's home page and each of the portlets might be very different on a mobile phone compared with those on a desktop browser.

Supporting new devices

Because WebSphere Portal supports several different markup languages, portlets can render themselves for a variety of desktop and mobile browsers. Portlets that do not natively support the device markup can optionally be transformed using transcoding technology. This means that portlets can easily and automatically support mobile devices, even if the portlet developer did not explicitly support that device. The framework for supporting markup languages is based on open standards and is extensible – making it easy to support other markup languages or new devices.

To support new browsers and devices, you can add new markup and clients using the corresponding administration portlets. In the markups portlet, the markup name indicates the name of the folders that are used to store the page templates and the theme or skin files matching that markup language (see Figure 56).

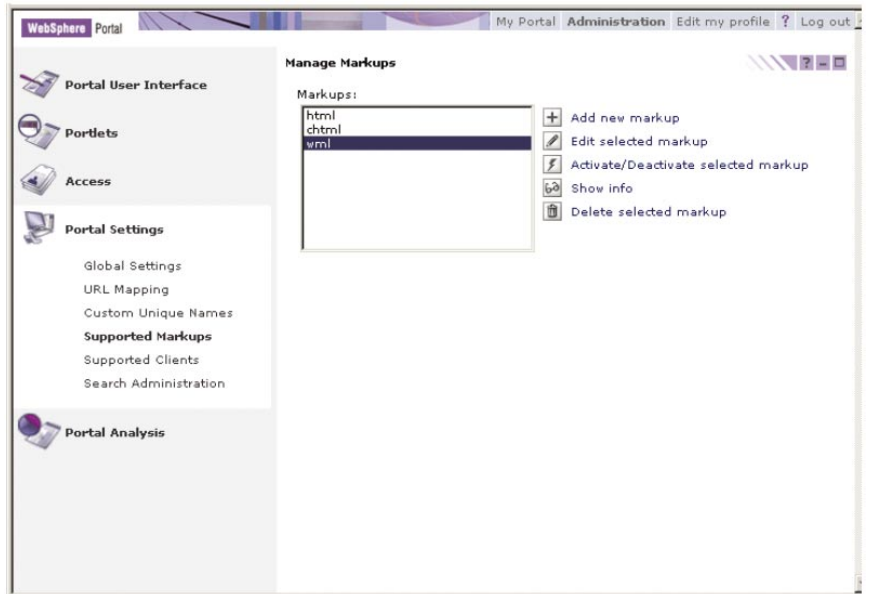


Figure 56. Enabling the portal for new markup languages

To add a markup, you must create a new entry specifying the Multipurpose Internet Mail Extensions (MIME) type and the character set associated with that markup. You also need to add all the JSP templates associated with supporting a markup, such as new layouts, panels, skins and style sheets. When the portal server receives an HTTP request, it matches the values in the user agent header against known patterns that identify common browsers for desktops, mobile phones and other devices. Entries for common clients are already set up, but you can add new ones using the manage clients portlet (see Figure 57).

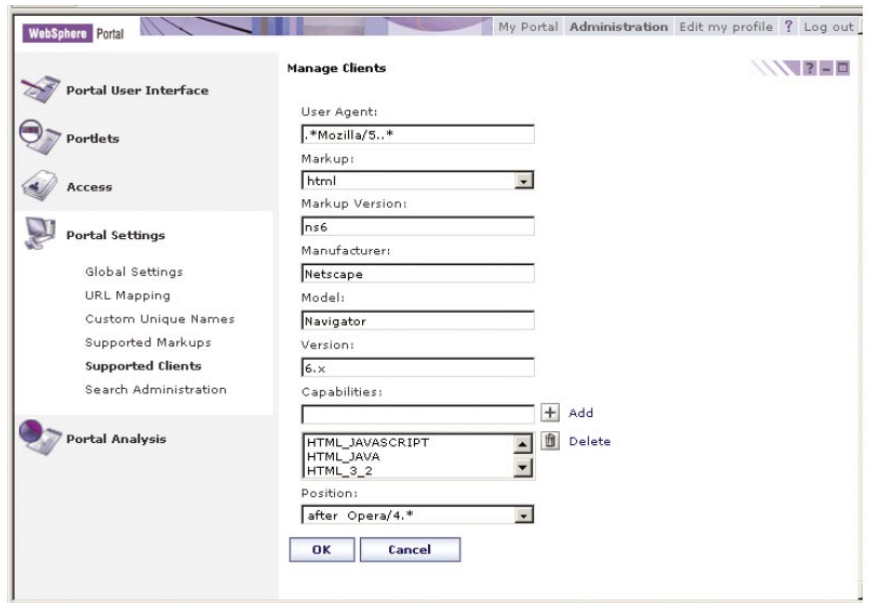


Figure 57. Enabling the portal for new client browsers

The order of the client entries determines the order in which the patterns are applied against incoming HTTP requests. For further information, visit the WebSphere Portal information center at ibm.com/developerworks/websphere/zones/portal/proddoc.html.

IBM provides other tools and products to enable more extensions of WebSphere Portal applications to mobile devices and protocols. IBM WebSphere Mobile Access is a middleware platform that provides an integrated client-server environment and application-development tools to extend business applications and data to mobile users operating in connected or occasionally connected modes. It enables deployment of business data and application services, including portal-based applications, on mobile devices and provides a back-end integration framework, including administration and management services. WebSphere Mobile Access also includes services for e-mail and PIM synchronization, Web access and notification services.

IBM WebSphere Everyplace[®] Mobile Portal provides a multichannel device-authoring system, device templates and specific support for extending portal applications to mobile devices, supporting users of mobile devices operating in a connected mode (see Figure 58). WebSphere Everyplace Mobile Portal supports the XML Device Independent Markup Extensions standard, enabling support to compare source content to specific mobile device interface attributes and write appropriate interfaces supporting one or multiple device types. This support enables you to preserve the unique interface benefits of specific mobile interfaces when working with WebSphere Portal source content.

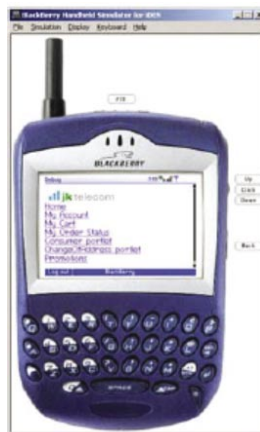


Figure 58. Interact with portal content through mobile-device interfaces optimized for device display by WebSphere Everyplace Mobile Portal.

A complete platform for collaboration and enterprise application integration

IBM WebSphere Portal offers an enterprise portal platform with the complete portal services necessary to deliver a single point of personalized interaction to applications, content, business processes and people. Its portal framework enables you to deliver more value by leveraging applications and data from previous investments, leading to immediate return on investment (ROI). Using WebSphere Portal, you can quickly build scalable portals to help simplify and improve the user's experience with personalized content, business processes and applications. And because the WebSphere Portal framework is built on open standards, you can increase and retain your business agility by easily adapting your portal solutions to support new requirements, positioning your portal infrastructure to capably manage current and future application- and integration-management requirements.

For more information

To learn more about IBM WebSphere Portal, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/websphere/portal



© Copyright IBM Corporation 2005

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
05-05
All Rights Reserved

DB2, DB2 Universal Database, Domino, IBM, the IBM logo, Everyplace, Informix, iNotes, Lotus, Lotus Notes, the On Demand Business logo, QuickPlace, Rational, Sametime, SmartSuite, Tivoli, WebSphere, Workplace and the Workplace family of marks, and z/OS are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

International Business Machines Corporation provides this paper "AS IS" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability or fitness for a particular purpose. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Information in this paper as to the availability of products (including portlets) was believed accurate as of the time of publication. IBM cannot guarantee that identified products (including portlets) will continue to be made available by their suppliers.

This information could include technical inaccuracies or typographical errors. Changes may be made periodically to the information herein; these changes may be incorporated in subsequent versions of the paper. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this paper at any time without notice.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to IBM Director of Licensing, IBM Corporation, 4205 South Miami Boulevard, Research Triangle Park, NC 27709, U.S.A.