

Table of contents

About the author	3
Introduction	4
Agenda.....	5
Today's diverse and evolving workplace	6
It is all about people	7
... into an integrated dynamic workplace	9
Portal, portlet and portal server defined	10
What is a portal?	11
Portal principles	12
Portal = integration at the glass	13
Portals defined	14
The many faces of a portal	15
Portals consist of several components	16
Another sample portal.....	17
What is a portlet?	18
What is a portal server?	19
IBM WebSphere Portal Server overview	20
WebSphere Portal extends the portal concept	21
WebSphere Portal demonstration sites	22
Magic quadrant: Portal leadership 2006	23
New with WebSphere Portal Version 6.....	24
WebSphere Portal V6 family on i5/OS.....	25
WebSphere Portal portfolio — visual summary	26
Functions not yet supported on i5/OS.....	28
Meeting minimum hardware requirements.....	29
Sizing the solution — be careful, be methodical	30
WebSphere Portal conceptual architecture	31
WebSphere Portal — the face of SOA.....	33
What is a service-oriented architecture?.....	34
WebSphere Portal — front end to SOA	35
SOA and WebSphere Portal add value	37
The value of WebSphere Portal and SOA	38
Portlet-application development.....	39
Developing portals and portlets	40
Portal and portlet development	41
Developing custom portlets.....	42
Portal applications.....	43
Portlet-application architecture	45
Portlet application architecture (continued).....	46
Portlet JSP example	47
Portlet API: IBM Portlet API or JSR 168	48
Why use the JSR standard?	49
Portlet API: Request processing	50
Portlet API: Customization	51
Portlet API — messages.....	52
Portlet API — events.....	53
Portlet API — two-phase processing	54
IBM Portlet API — messaging	55

IBM Portlet API — click-to-action	56
IBM Portlet API messaging and C2A	57
A sample application — portlets in action	58
Portlet API — multiple-device support	59
Portlet API — themes	60
Integrating with System i applications and data	61
Integrating with System i	62
Other portlet builders and purchased portlet applications.....	63
WebSphere Portal Business Solutions Catalog	64
Portlet-builder tools	65
Making portlets from existing Web applications.....	66
Web applications and portal applications.....	67
Portalizing from the ground up	68
Portal-application deployment.....	69
Summary.....	70
Resources.....	71
Trademarks and special notices	73

About the author

Brian Golish is a technical consultant with the IBM ISV Solutions Enablement team in Rochester, Minnesota. Brian has a strong background in software development and architecture and has worked with many IBM Business Partners to incorporate IBM Lotus technologies into their solutions. His focus is primarily on application development with Lotus technologies, such as: Lotus Domino, Lotus Notes, IBM Sametime, IBM QuickPlace and IBM WebSphere Portal. Brian joined IBM in 1998 and has since developed and deployed several successful tooling projects for the IBM OS/400 and IBM i5/OS development laboratory. He joined the ISV Solutions Enablement team in September of 2005.



Developing a portal interface

Brian Golish, IBM Corporation
ISV Technical Enablement – Lotus
bgolish@us.ibm.com



© 2007 IBM Corporation

Introduction

Welcome to this online course entitled “Developing a portal interface.”

Portal technology is one of the fastest-growing areas in the IT industry today.

This online course provides architects and application developers with basic knowledge of portals and portlets and provides information on the IBM® tooling that is available to design, develop and deploy a portal architecture. The course content also includes an in-depth look at the Java™ (JSR168) and IBM Portlet APIs and the function and features provided by each. Finally, the presentation discusses how the IBM WebSphere® Portal allows you to move into a service-oriented architecture (SOA).

Audience: This course is intended for programmers and nonprogrammers who need to build portlets that display application-based data in WebSphere Portal. This includes IBM clients, independent software vendors (ISVs), systems integrators and IBM technical staff. Architects and developers responsible for implementing solutions using WebSphere Portal V6 will benefit, as well.

Prerequisites: Students should have a basic understanding of Web-application (servlet) development concepts.

Agenda

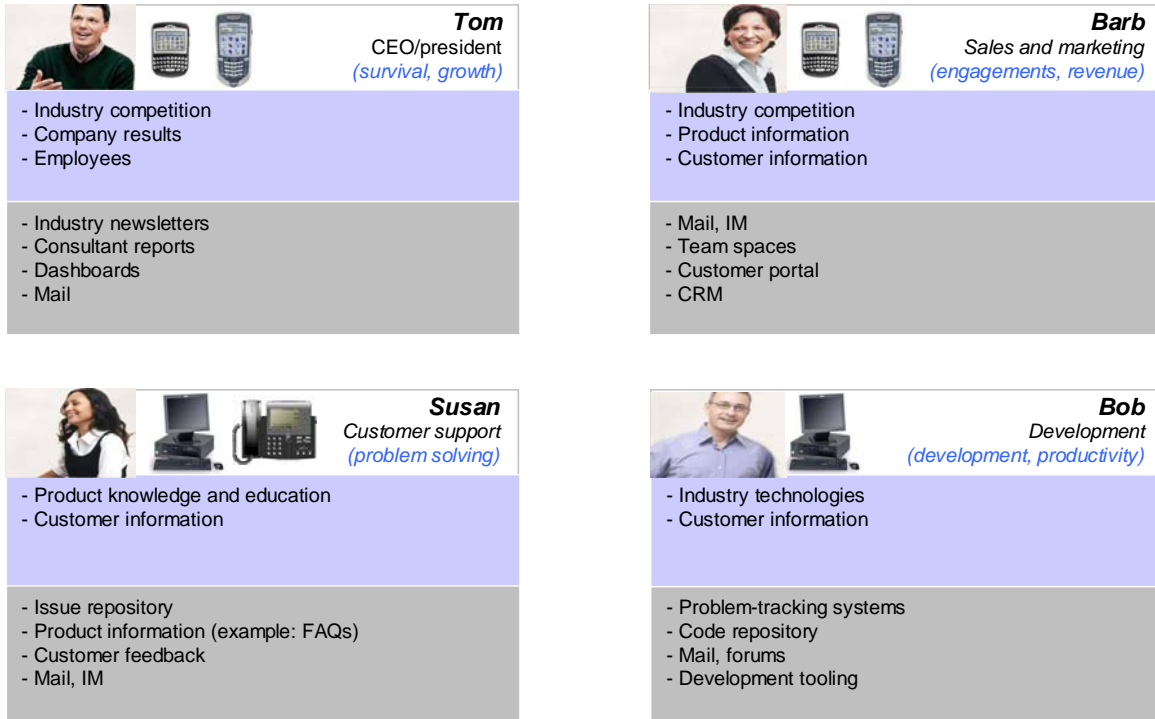
- **Portal, portlet and portal server defined**
- **WebSphere Portal Server overview**
- **WebSphere Portal - the face of SOA**
- **Portlet-application development**

Agenda

This course will cover the following subject matters:

1. Give a brief overview of basic portal, portlet and portal server concepts.
2. Present the IBM WebSphere Portal Server offerings that are available for the IBM i5/OS® operating system, which runs on the IBM System i™ platform.
3. Define how IBM WebSphere Portal can act as an entry point to SOA.
4. Describe the details involving portlet development, including the Java (JSR168) and IBM Portlet APIs. This discussion often times will distinguish between Web-application (servlet) development and portlet development.

Today's diverse and evolving workplace



Today's diverse and evolving workplace

This is a diagram of a typical business-organization situation.

Even though this represents an IT company, the concerns represented here are relevant to most other industries.

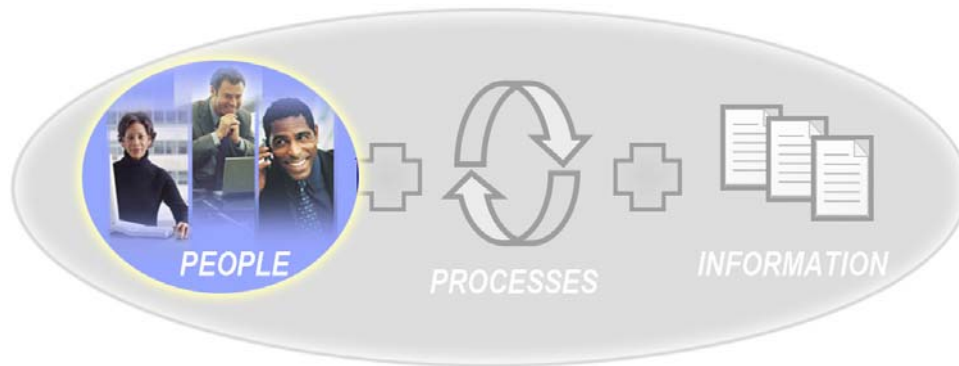
Depending on an individual's role within an organization, that person needs access to certain data, business processes and tools.

Making the situation complex is the fact that it is also necessary for individuals to communicate and collaborate with other people who have different styles and preferences for viewing data and using the business tools.

Complicating the situation further is the fact that information is managed, presented and shared through multiple user interfaces and from multiple repositories.

It is all about people

An On Demand Business is an enterprise whose business can respond in real time to any customer demand, market opportunity or external threat.

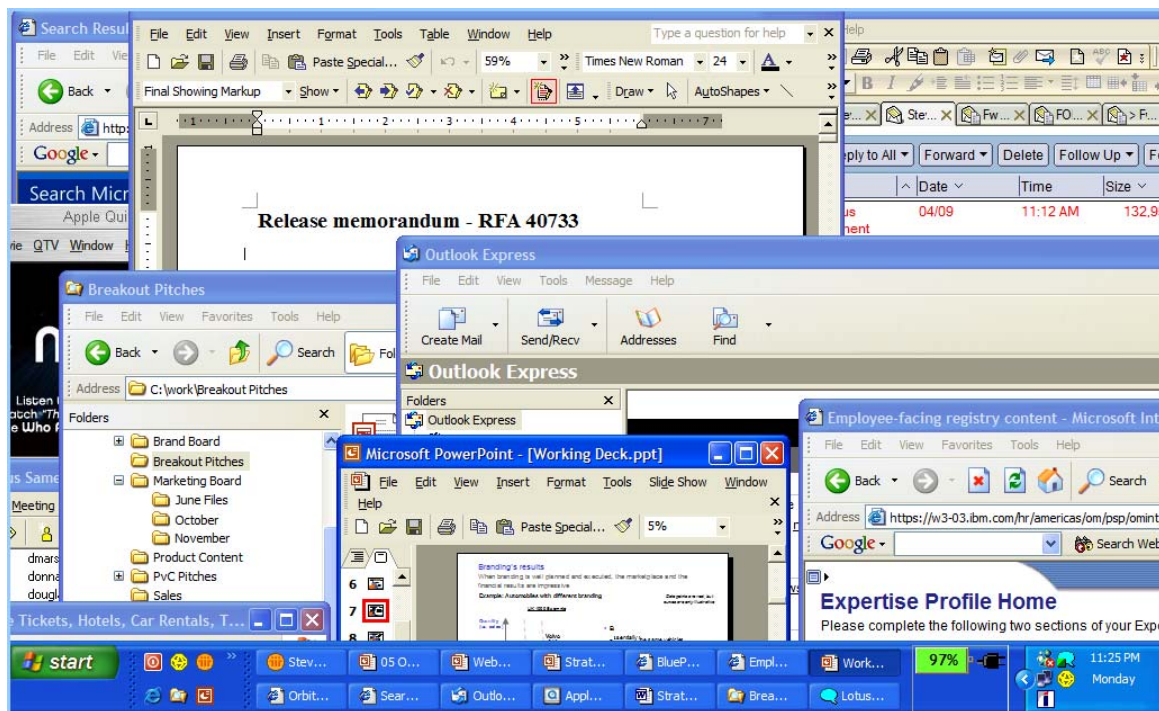


It is all about people

People, processes and information are important in an On Demand Business. Business processes need to be well defined and efficient in their methodology. Additionally, the data must be accurate and consistent.

The “people” component is extremely important when defining processes. After all, people are the resource that monitors and reacts to the organization’s business processes and information. It is the primary goal of the processes to make the individuals who use them highly efficient and productive.

Evolving today's disparate user experiences...

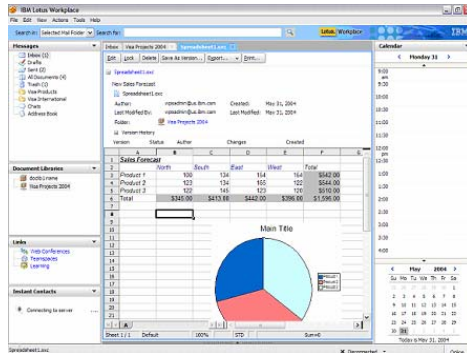


Evolving today's disparate user experiences ...

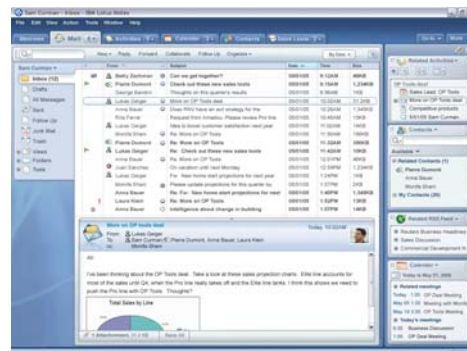
This is a common experience among many business organizations and companies today. This complexity comes at a cost. Users are not able to be as flexible and efficient as they could and should be. Furthermore, the cost to manage and administer these complex environments is high and becomes even more costly as your company and IT environment grow.

...into an integrated dynamic workplace

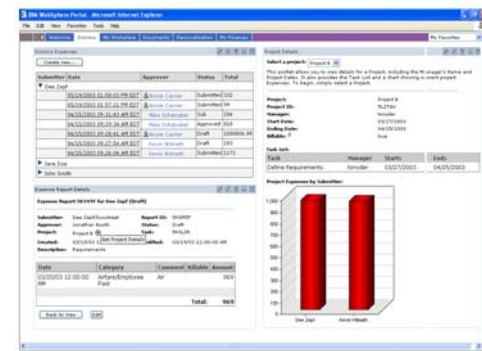
Lotus Expeditor



IBM WorkPlace



Lotus Notes



WebSphere Portal

... into an integrated dynamic workplace

IBM is committed to making it easier to evolve these disparate user experiences and business challenges into an integrated dynamic workplace.

In particular, the IBM portal technologies and their related portlet-development tools make it easier for developers to focus on a strategy for integrating many user experiences through a single, consistent point of access.

Agenda

- ✓ **Portal, portlet and portal server defined**
 - WebSphere Portal Server overview
 - WebSphere Portal - the face of SOA
 - Portlet-application development

Portal, portlet and portal server defined

The next section of this course discusses some basic concepts involving portal, portlets and portal servers.

What is a portal?

A portal is a Web interface that offers a single point of personalized access to applications, content and processes.



What is a portal?

Portals aggregate content and integrate applications, processes and human beings. In effect, an enterprise portal provides human integration – connecting a user to all the information resources that are necessary to do business with a company, work for a company or interact with a government.

Beware of developing applications for portals that focus on just one type of access. A robust, fully functional portal provides access to content, applications, processes and human resources, for example. An enterprise portal that is built with a narrow focus is likely to need expansion within 18 months.

Action item: Part of a successful portal project is to create a shared vision of the portal's uses and benefits for your organization. If there is no need to create relevancy for differing user audiences, do not spend the time and money to build a portal – build a Web site instead.

Portal principles

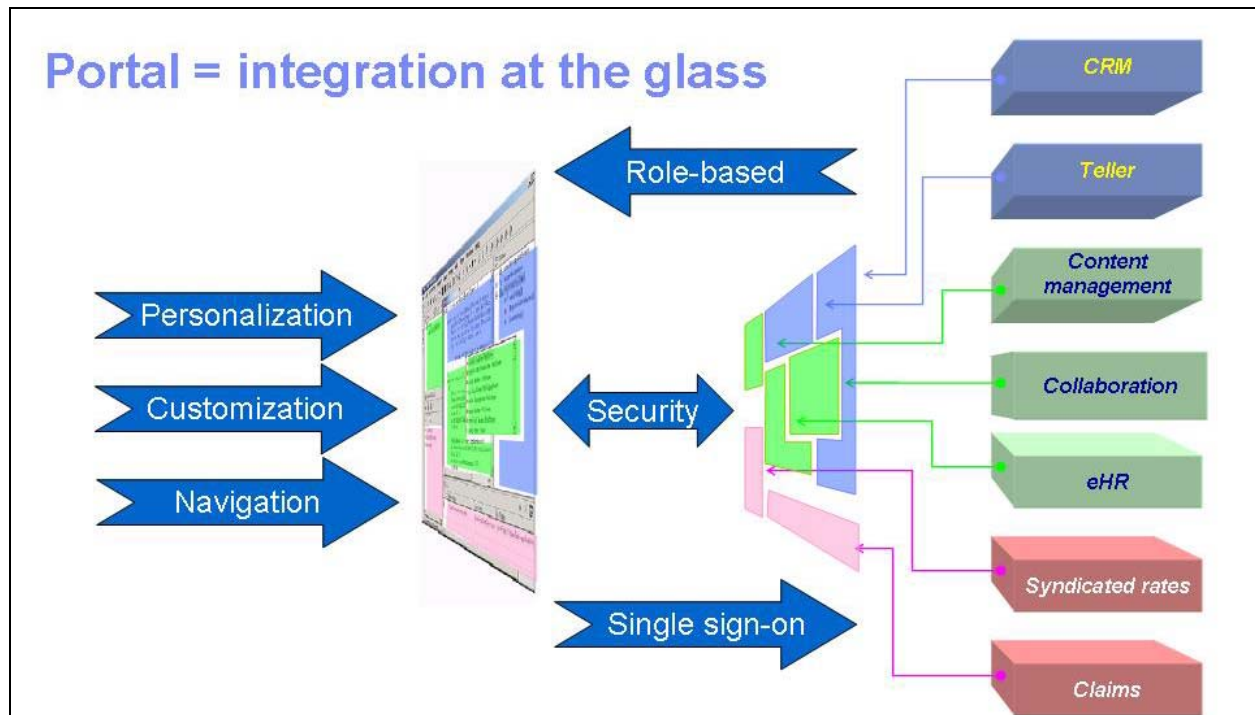
- Combines portlets into a unified presentation
- Delivers a personalized experience that considers role, personal settings, administration and device settings
- Separates site design, assembly and administration from application design
- Provides application integration



Portal principles

Consider these four principles when determining whether a portal offers value for your organization.

- A portal combines portlets into a single, unified presentation.
- A portal delivers a highly personalized experience for each user. The portal is designed to consider the user's role, as well as personal and device settings. The portal also takes into account settings that the administrator has established regarding what information and applications are to be made available to the user.
- A portal separates site design, assembly and administration from the application-design process for each portlet that the user will be capable of accessing from the portal.
- A portal provides a systematic method for the integration of applications.



Portal = integration at the glass

This diagram illustrates the value offered by a portal as compared to the benefits of Web pages.

The boxes on the right represent the different applications and content that exist in a company. They are color-coded to show that they are coming from one department or business unit.

In the middle, the portal assembles the various pieces (for example, pages, portlets and rights) based on considerations that are related to the user's role, security authorization and personalization preferences.

On the left, you can see that the resulting assembled portal displays an integrated experience of relevant applications and functions for the user.

Portals defined

■ Classes of portals

- Megaportals and enterprise portals
- Vertical and horizontal portals
- Internal facing and external facing

■ An enterprise portal is...

... a Web software infrastructure that provides access to and interaction with **relevant** information, content, applications, business processes and Human resources for select, **targeted** audiences – delivered in a highly **personalized** manner.



Portals defined

Portals have been in existence since the mid-1990s, and portal products have been available since 1997. Despite their popularity, the term *portal* continues to be one of the most misunderstood and abused terms in the IT industry. *Megaportals*, such as Yahoo, continue to be better understood on the consumer side, but enterprise portals are the greater focus for most companies.

A key aspect of a portal is its relevancy: A portal that displays the same information for all users is not really a portal — it is merely a Web site, regardless of its level of sophistication. In contrast, a portal provides relevancy to each individual user through its use of personalization and customization functions. These features allow the administrator to target the content of the portal page to potentially very small audiences, even down to an audience of one. This one-to-one customization is actually critical when a portal is deployed in a marketing context in a business-to-consumer (B2C) environment.

The many faces of a portal

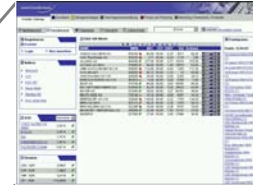
In-side (B2E) experience

- Enterprise-wide communications
- Common-management platform of selected corporate functions
- Branding your way of doing business
- Information management and access



Buy-side (B2B) experience

- Collaborative management of supply-chain transactions
- Personalized partner relationships
- Integrated with supply-chain applications and processes



Sell-side (B2C) experience

- Customer-services brand management
- Personalized customer marketing and sales
- Multichannel sales integration
- Web-based customer-service delivery



The many faces of a portal

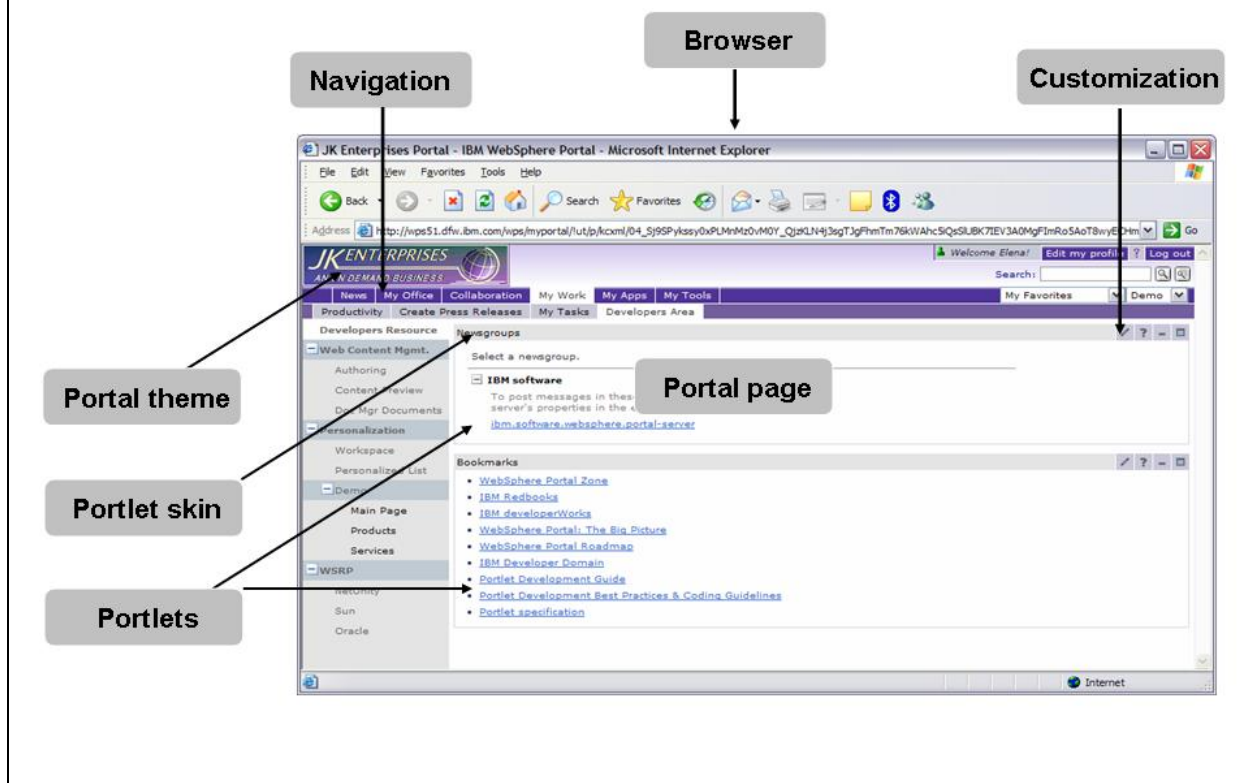
An enterprise portal generally focuses on three types of experiences.

Buy-side (B2B) experience: Functions related to business-to-business (B2B) include provisions for the collaborative management of supply-chain transactions. These portals also provide capabilities for personalizing the relationship with the enterprise's business partners. For this reason, B2B portals are integrated with supply-chain applications and processes (purchasing, accounts payable, inventory, customer-ordering habits and other similar functions).

In-side (B2E) experience: Functions related to business-to-enterprise (B2E) facilitate enterprise-wide communications. The portal can become the common management platform of selected corporate functions. At the same time, these portals make it easy to enforce consistent branding of the way in which the enterprise conducts business. These portals provide access to information, as well as a formalized means of managing the data by authorized users.

Sell-side (B2C) experience: Functions related to B2C support the management of the customer-services brand. This includes personalized customer marketing efforts and sales mechanisms. It is typical for B2C portals to integrate multichannel sales and deliver Web-based customer service.

Portals consist of several components



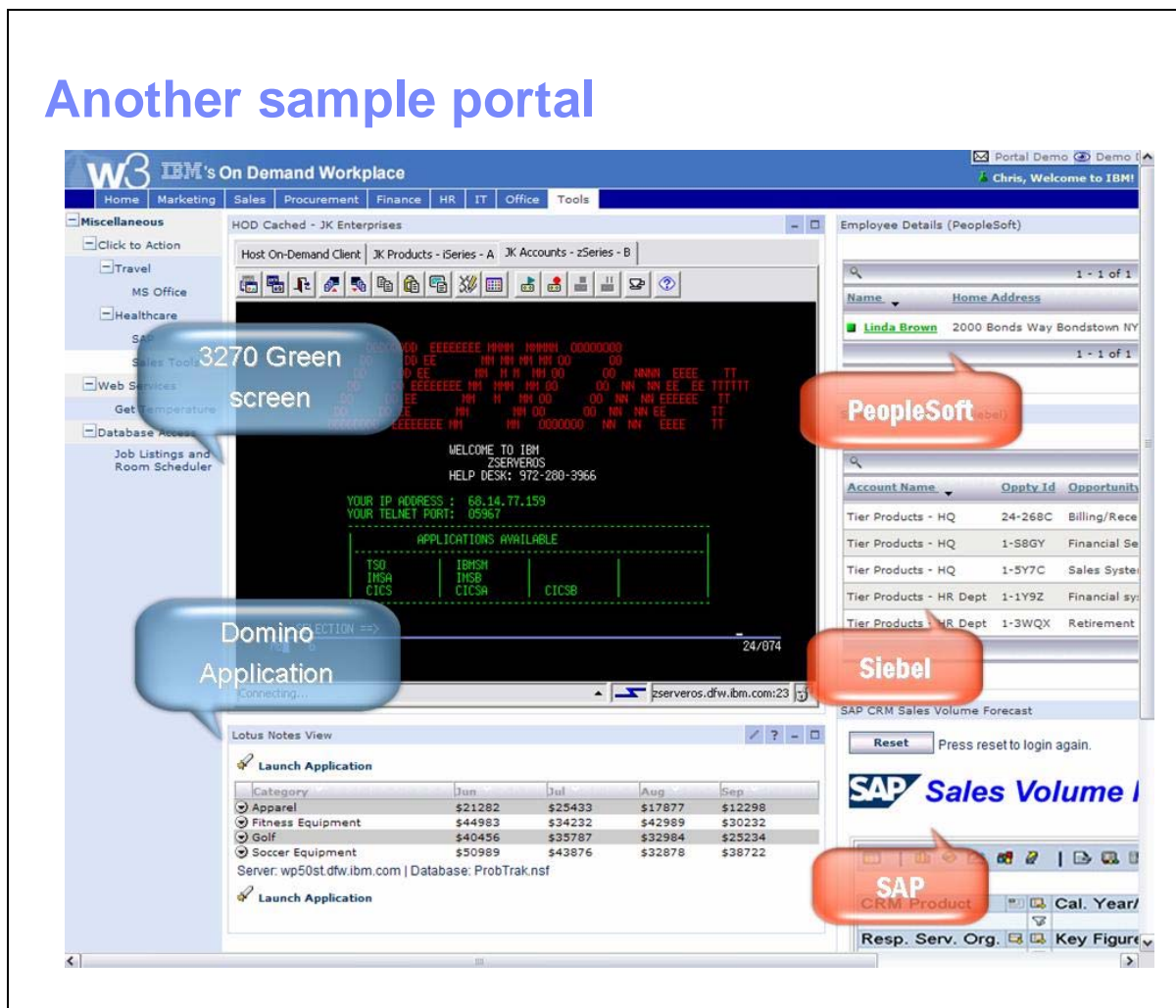
Portals consist of several components

This screen capture shows a sample enterprise portal. As mentioned earlier, a portal is a Web application that you can access, just as you would access any other Web application — through a Web browser.

At first glance, a portal might look like a regular Web application. But it is much more than that. Here are some hints that make it easier to recognize a portal:

- Most portals have a theme. A theme consists of a header that contains information about a portal. Prepackaged portal offerings usually provide several sample themes. However, these themes are usually easy to modify; or you can create custom themes.
- A portal also has multiple windows, which are called *portlets*. As the user begins to interact with a portal, the various portlets that appear on the site can send information to each other.
- A portal also provides easy and intuitive navigation between portal pages.

Another sample portal



Another sample portal

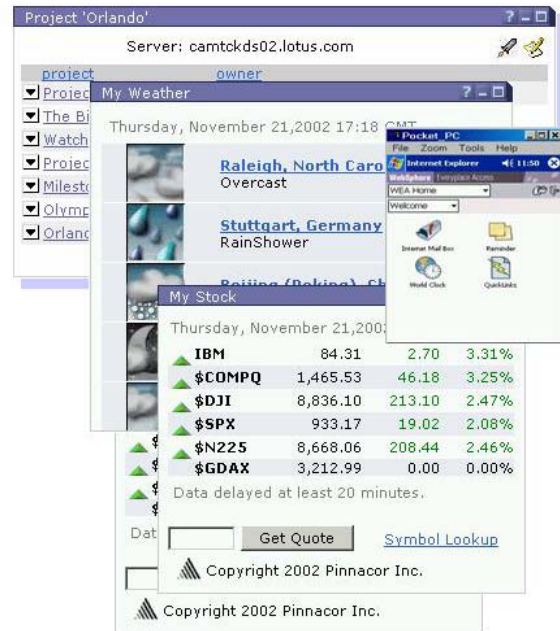
Here is an example of a different portal site.

Notice that it has a navigation bar on the left. It also displays a green-screen application, as well as an IBM Lotus® Domino® application in the bottom center of the page. The right side of the portal display details related to several IBM Business Partners.

The user of this portal has everything necessary to multitask. This includes collaborative efforts, monitoring of partner information and using a TN3270 application.

What is a portlet?

- **A portlet is a pluggable component that is hosted by the portal.**
- **Each portlet is a separate application.**
 - Acts as a front end to a larger application
 - Can be placed anywhere on the page
 - Can be dynamically deployed and configured
- **Portlets can support multiple devices.**
 - Provides output to the user by generating markup that is assembled into a portal page by the portal
 - Can run on cell phones, organizers
 - Can use voice-activation technologies
 - Provides unique views for each device
 - Can share business logic



What is a portlet?

A portlet is an application component that can be plugged into the portal that will serve as the host for that application.

Every portlet is a separate application that acts as a front end to a larger application. You can place portlets anywhere on the portal page. It is also possible to dynamically deploy and configure portlets.

Portlets can be quite versatile in the devices that you allow to access them. This is possible because the portlet generates XML and other markup components that the portal assembles for the user. The portal makes a decision of how to put the markup data together depending on the particular device that the user is using to access the portal. A well-designed portal can display unique views of the portlet applications on cell phones and personal digital assistants (PDAs). It can even support voice-activation technologies to interact with the user as an additional means of sharing business logic.

What is a portal server?

- **Provides common services**
 - Application connectivity
 - Integration
 - Administration
 - Presentation
- **Serves portlets to the user**
- **Provides features that relieve developers of coding tasks**

What is a portal server?

A portal server provides common services for both the user and the portlet applications.

It provides application connectivity, integration, administration services and presentation tasks.

Functionally, a portal server serves the portlets to the user. It also provides many features that developers must otherwise write themselves. For example, if there is no portal server, you have to develop your own servlets. Thus, portal servers support portlet development that is much more cost-effective.

Agenda

- Portal, portlet and portal server defined
- ✓ **WebSphere Portal Server overview**
- WebSphere Portal - the face of SOA
- Portlet-application development

IBM WebSphere Portal Server overview

The next section of this course examines the IBM WebSphere® Portal Server offering and how it extends the concept of a portal server.

WebSphere Portal extends the portal concept

- **Middleware that creates and runs portals**
- **Provides support for many functions**
 - Application and information integration
 - Customization and personalization
 - Single sign-on
 - Document and content management
 - Collaboration functions
 - Instant messaging
 - Web meetings
 - E-mail
 - Extensions to pervasive devices (transcoding)
 - Internationalization
- **Building and integrating business applications is still the developer's responsibility**

WebSphere Portal extends the portal concept

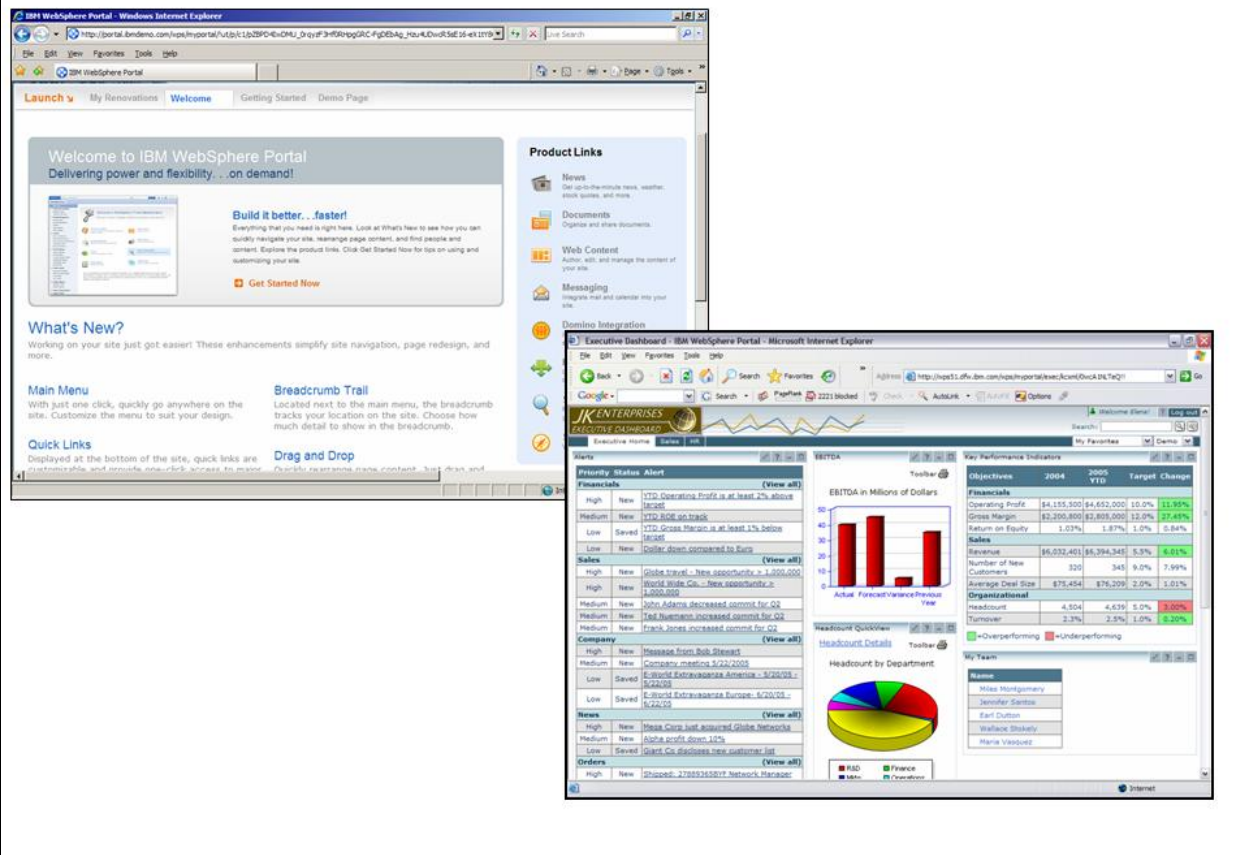
IBM WebSphere Portal is middleware that makes it easier for you to create and run portals.

WebSphere Portal provides support for:

- Application and information integration
- Customization and personalization
- Single sign-on
- Document and content management
- Collaboration functions (including instant messaging, Web meetings and e-mail tasks)
- Extension to pervasive devices (transcoding or reformatting application data for PDAs and cell phones)
- Internationalization

WebSphere Portal is a versatile and robust offering, but keep in mind that it is your job, as a developer, to build and integrate the applications that are deployed to WebSphere Portal because WebSphere Portal does not automatically do these things.

WebSphere Portal demo sites

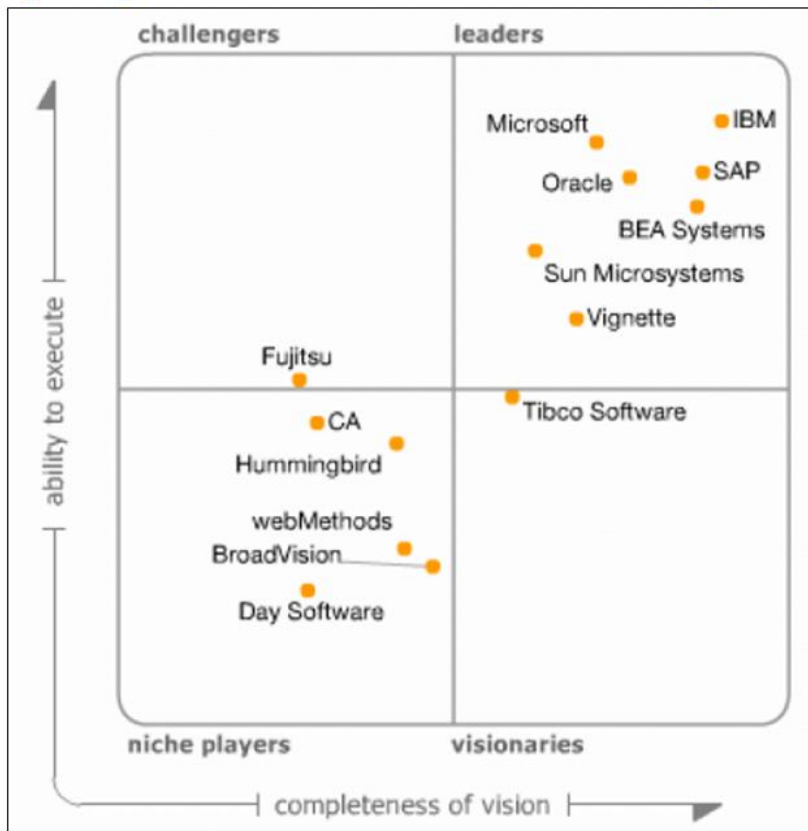


WebSphere Portal demonstration sites

IBM provides two Web sites where developers can go to see a demonstration of the WebSphere Portal offering. There are as follows:

- The WebSphere 5.1 Portal demonstration site:
www.wps51.dfw.ibm.com/wps/portal/
- The WebSphere 6.0 Portal demonstration site (requires cost-free registration):
<http://portal.ibm.demo.com/wps/portal/>

Magic quadrant: Portal leadership 2006



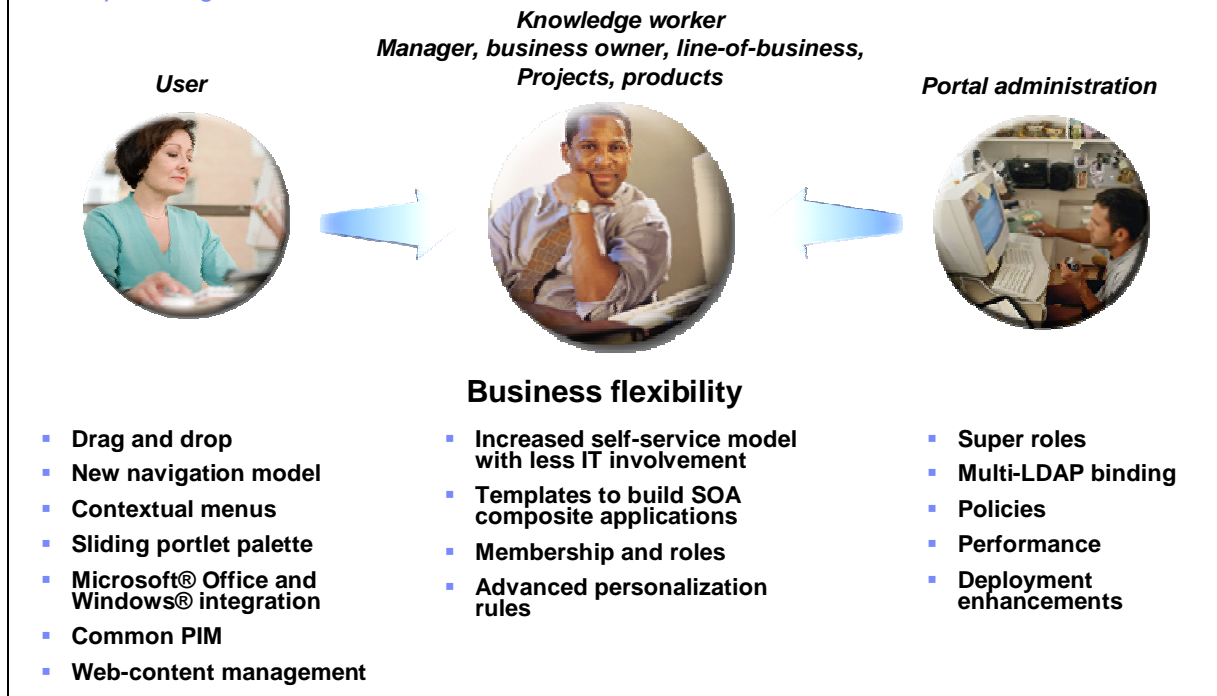
Magic quadrant: Portal leadership 2006

This Gartner study shows that WebSphere Portal is a market leader in its segment.

This Magic Quadrant (copyrighted by Gartner, Inc. on May 16, 2006) is a graphical representation of a marketplace at and for a specific time period. It depicts a Gartner analysis of how certain vendors measure against criteria for that marketplace, as defined by Gartner. Gartner does not endorse any vendor, product or service depicted in the Magic Quadrant and does not advise technology users to select only those vendors placed in the Leaders quadrant. The Magic Quadrant is intended solely as a research tool and is not meant to be a specific guide to action. Gartner disclaims all warranties, express or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

New with WebSphere Portal V6

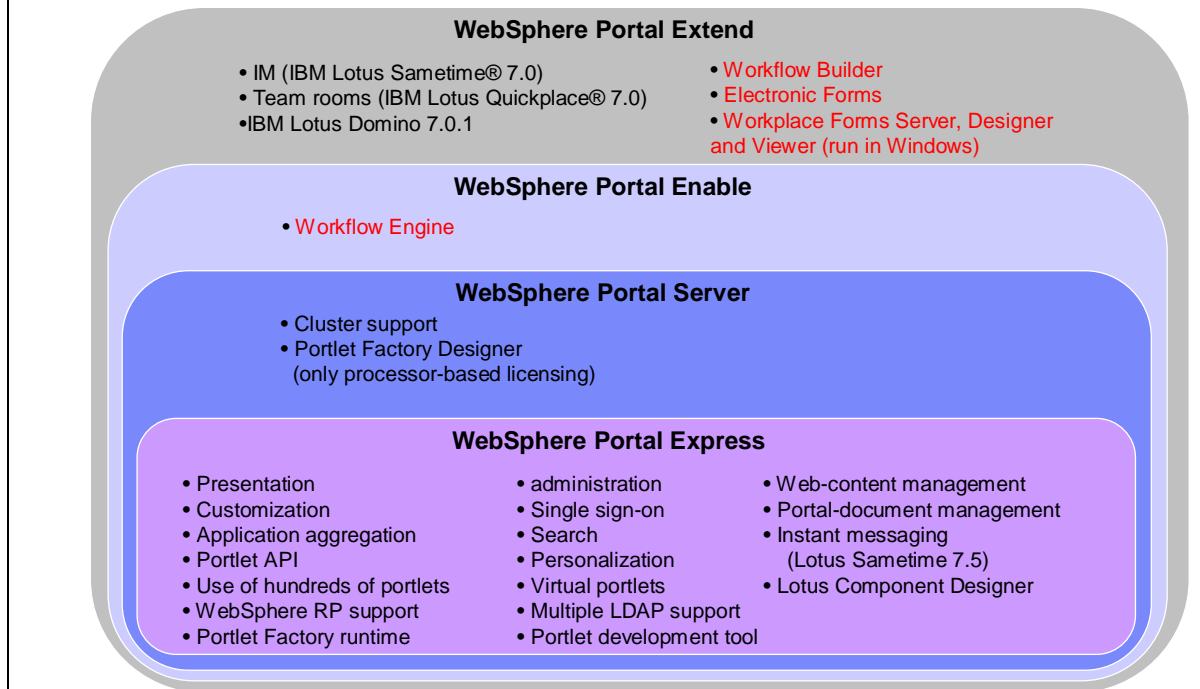
Empowering the business user



New with WebSphere Portal Version 6

The latest release of WebSphere Portal is Version 6. This chart shows a short list of the enhancements provided with this release.

WebSphere Portal V6 family on i5/OS



WebSphere Portal V6 family on i5/OS

The IBM i5/OS operating system supports the entire WebSphere Portal V6.0 product family. This includes the following:

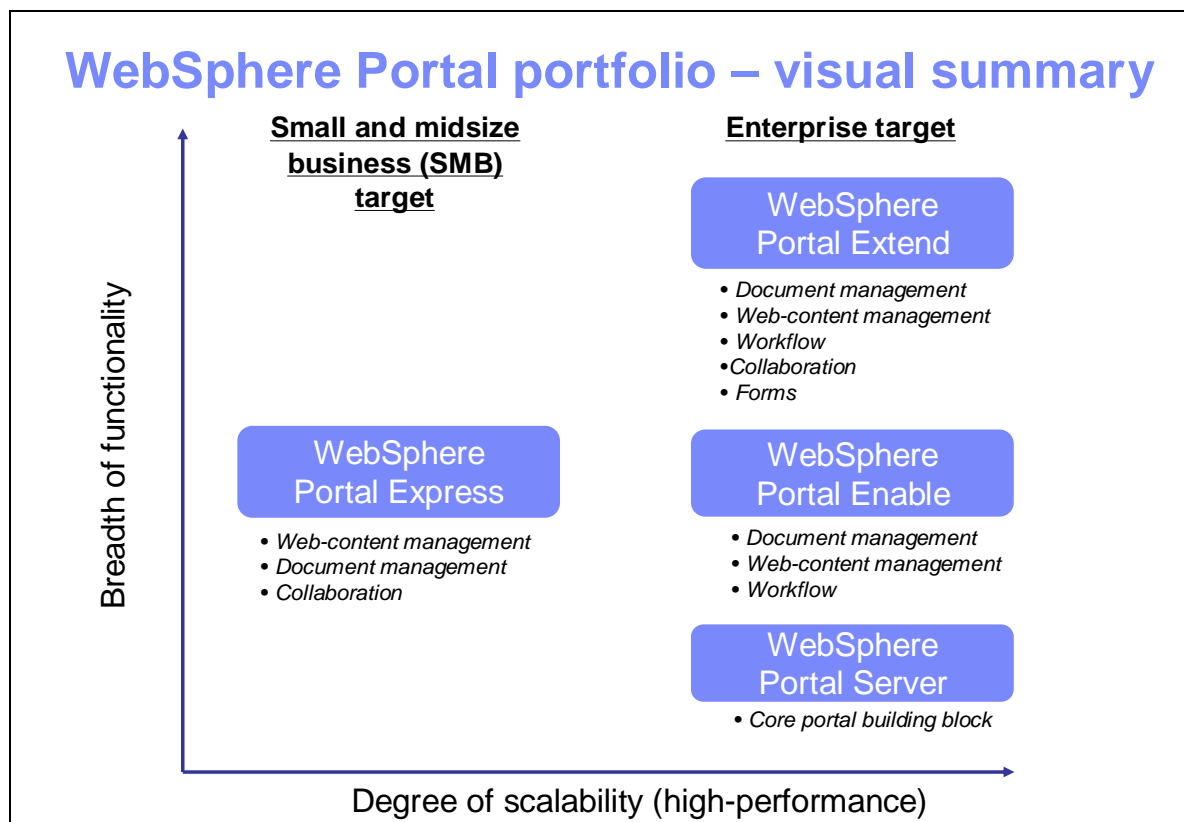
- IBM WebSphere Portal Extend
- IBM WebSphere Portal Enable
- IBM WebSphere Portal Server
- IBM WebSphere Portal Express

However, there are some components of these products that i5/OS does not yet support. For instance, IBM Workflow Engine (which is a part of WebSphere Portal Enable) and IBM Workflow Builder (which is part of WebSphere Portal Extend) are not yet supported because IBM WebSphere Process Server is not yet supported on the System i platform. (**Note:** These products are explained in a later slide.)

i5/OS also does not support Electronic Forms (which is a part of WebSphere Portal Extend) because Workplace Forms does not run on the IBM System i platform. However, the IBM Workplace Forms™ components, IBM Workplace Forms Viewer, IBM Workplace Server and IBM Workplace Designer are included in the WebSphere Portal V6.0 for i5/OS package. You can run these components on a Microsoft Windows server.

Visit the following Web site to find details about these offerings and their features:
ibm.com/software/genservers/portal/features

There is also a link to this Web site in the References section of this course.



WebSphere Portal portfolio — visual summary

This chart is a visual perspective of the market positioning for the four WebSphere Portal products.

WebSphere Portal Express is designed for the small- and midsize-business (SMB) market. It provides three fundamental sets of functions that are typically most important to organizations of this size, including Web-content and document management as well as collaboration support. This product helps businesses aggregate information, as well as create and manage Web sites (including intranets, extranets and the Internet). WebSphere Portal Express also makes it possible to store and collaborate on documents. Additionally, there are functions for presence awareness and instant messaging. This product is installed on a single server; it is easy to deploy and provides prebuilt Web sites for rapid implementation.

WebSphere Portal Server is the core portal building-block offering for larger enterprises. It helps the business to aggregate information and can be deployed on multiple servers. This is a high-performance product that lets the company scale up to more sophisticated portal sites as future needs dictate. This product is ideal for enterprises that already have an existing collaboration infrastructure.

WebSphere Portal Enable includes the core enterprise portal and provides document-management, Web-content management and workflow tools. In addition to aggregating information, this product makes it possible for the business to store and manage documents as well as manage complex workflows. It is also possible to create and manage Web sites. This high-performance offering is highly scalable and can be deployed to multiple servers. Enterprises that select this product also already have an existing collaboration infrastructure.

WebSphere Portal Extend provides all the functionality of WebSphere Portal Enable and, additionally, includes sophisticated collaboration tools (for presence awareness and instant messaging) and electronic forms to support the more complex team- and people-interaction needs of large enterprises. Deployable to multiple servers and highly scalable, this offering is positioned to solve the needs of enterprises when implementing a collaborative infrastructure.

Functions not yet supported on i5/OS

- **WebSphere Process Server is not supported on i5/OS**
 - Workflow Builder and Workflow Engine are not supported
- **Electronic Forms is not supported**
 - WebSphere Portal and Workplace Forms Integrator are not supported
- **WebSphere IP V6 is not supported**
- **32-bit JVM is not supported**
- **WebSphere Portal database support**
 - IBM Cloudscape™ and IBM DB2 for i5/OS are supported
 - All other databases are *not* supported
- **WebSphere Application Server is not installed by the WebSphere Portal installation process**
 - Installing WebSphere Application Server product is a prerequisite to installing the WebSphere Portal product
- **No WebSphere Application Server Install Factory support on WebSphere Application Server V6.0**
 - WebSphere Application Server iFixes must be installed manually

Functions not yet supported on i5/OS

WebSphere Process Server is not supported yet on i5/OS and, therefore, some Workflow functions that are available in WebSphere Portal Enable and WebSphere Portal Extend are not supported on i5/OS.

WebSphere Application Server Install Factory is not supported, which means that you must install iFixes manually.

Workplace Forms is not supported on i5/OS but System i users can install it in a Microsoft Windows environment that runs on an IBM Integrated xSeries® Server (IXS) for iSeries™ or IBM Integrated xSeries Adapter (IXA) for iSeries.

IBM WebSphere IP V6 is not supported.

There is no i5/OS support for the 32-bit Java virtual machine (JVM). However, even when i5/OS supported both 64- and 32-bit JVMs, WebSphere Portal V6.0 only supported the 64-bit JVM.

The only database supported by i5/OS is IBM DB2® for i5/OS, installed either locally or remotely. This point is important because it means that, even in Linux® or IBM AIX® partitions on the System i platform, the only database supported is DB2 for i5/OS.

Meeting minimum hardware requirements

- **Minimum recommended hardware for a production environment:**
 - 3800 CPW, 6 GB of memory, six to eight disk arms
 - At least one full processor (no partial processor)
 - L3 cache
 - For example, 1-way 1.9 GHz IBM POWER5™ processor
 - * Minimum was increased because of experiences with installation and configuration performance
 - * Reinforced by lab experiments on memory- and disk-constrained systems
- **No tuning results in poor performance**
 - Best Practices for WebSphere Portal V6.0 on i5/OS deployments
 - IBM WebSphere Portal Tuning Guide

Meeting minimum hardware requirements

This chart shows the minimum recommended System i hardware for a WebSphere Portal production environment. These minimums have been increased because of experience in the field with installation and configuration performance. These minimums are reinforced by lab experiments on memory- and disk-constrained System i models.

See the *IBM WebSphere Portal Tuning Guide for iSeries* for additional information. This guide is available at ibm.com/servers/enable/site/education/abstracts/wsptuning_abs.html.

There is a link to this Web site in the Links section of this course.

Hardware sizing – be careful, be methodical

- Take time to understand the customer's environment and to provide good input to the sizing tools
 - Plan for a staging or test environment with similar characteristics (topology) to the production environment
 - Use IBM Systems Workload Estimator for a rough hardware estimate
 - <http://www-912.ibm.com/wle/EstimatorServlet>
 - Consult with IBM Techline for formal proposals (IBM intranet):
 - EMEA Techline: <http://w3.ibm.com/support/emea/techline/sizing.html>
 - Americas Techline: <http://w3.ibm.com/support/americas/techline/sizewise.html>
 - For Business Partners: ibm.com/partnerworld/techline
 - System i Benchmark Center
- * **Important: Sizing does not replace capacity planning and stress testing.**

Sizing the solution — be careful, be methodical

You can also use the IBM Systems Workload Estimator to gain a rough hardware estimate for your WebSphere Portal installation. See the following Web site:
www-912.ibm.com/wle/EstimatorServlet

For IBM Business Partner members, the IBM Techline can help with your formal proposals. Visit the IBM intranet that is most appropriate to your geography:

EMEA Techline: <http://w3.ibm.com/support/emea/techline/sizing.html>

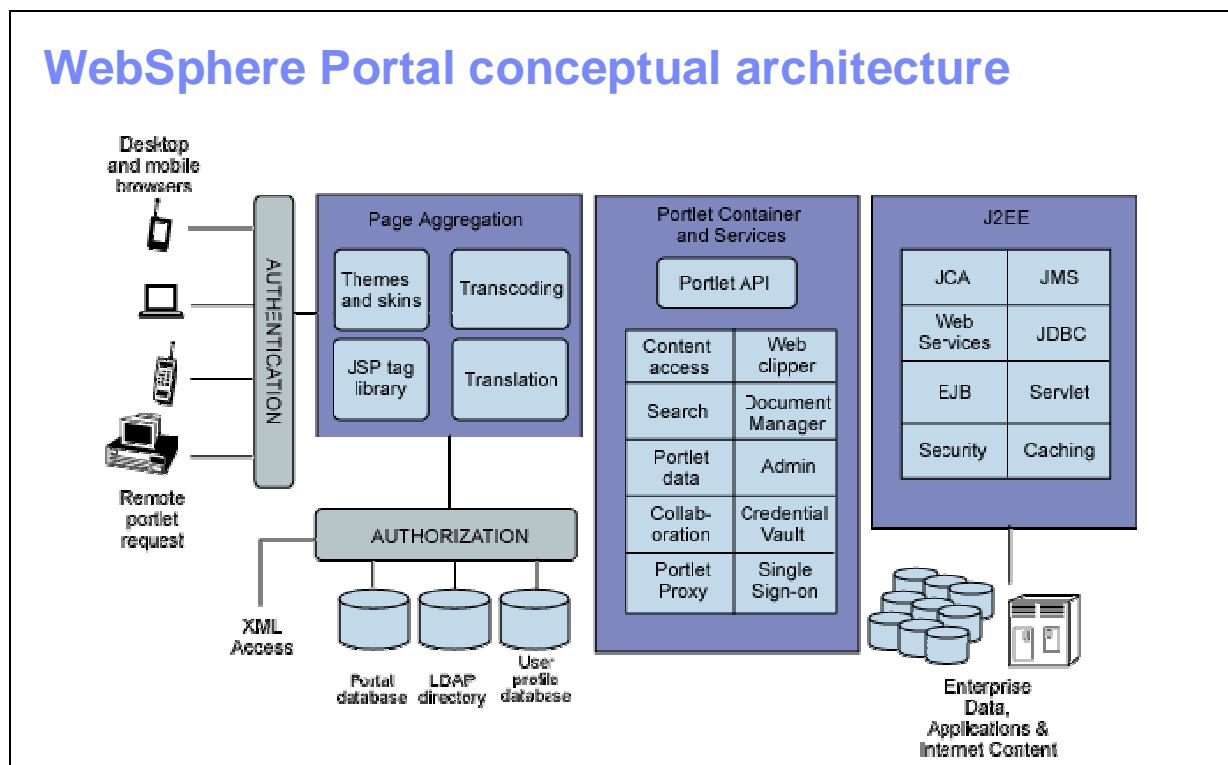
Americas Techline: <http://w3.ibm.com/support/americas/techline/sizewise.html>

For Business Partners: ibm.com/partnerworld/techline

You can also get the IBM System i Benchmark Center involved in helping you size your portal application to the most appropriate hardware configuration. Visit the following Web site:
ibm.com/servers/eServer/iSeries/benchmark/CBC/index.html

Links to these Web sites are available in the Links section of this course.

WebSphere Portal conceptual architecture



WebSphere Portal conceptual architecture

WebSphere Portal is rich in services. Trying to detail every service is beyond the scope of this course. However, it is helpful to address some of the aspects of WebSphere Portal by looking at the conceptual architecture, as shown here.

Page aggregation: To provide users with a consistent, customized view of portal applications on various client types, WebSphere Portal has services that can detect information about the user, the client device and the preferred language. Services can also determine which applications are accessible by the user. It is possible to combine the portlets into a usable view for the targeted client device

The portal's view is delivered in a branded and customizable manner using themes and skins. *Themes* determine the navigation controls; *skins* address the borders immediately surrounding each application snippet or portlet. As part of the aggregation process, WebSphere Portal determines and applies a user's preferences when rendering views. Transcoding services and a JavaServer Page™ (JSP) tag library support view rendering for multiple client devices. WebSphere Portal provides services for detecting the user's preferred language and delivering the appropriate translated content.

Portlet container and services: WebSphere Portal recently added support for the Java Portlet API, JSR 168. Services specified by JSR 168 include the following services: configuration, URL rewriting, window modes, portlet and portal contexts, request and response handling, session handling, servlet integration, security and view rendering. An important aspect of the Java Portlet Specification is the *cooperative portlet* services that enable a portal to provide some of the integration itself, which is referred to as *integration on the glass*. Portlets can use these services to share data and provide the user with a seamless experience.

For rapid application integration with the on demand workplace, you can use Web-clipping services to expose pieces of existing Web applications through the portal interface. IBM WebSphere Portal Document Manager provides an integrated solution for managing the ever-growing flood of documents. WebSphere Portal includes an integrated search engine.

WebSphere Portal provides an extensive series of portlets for administering the portal environment. WebSphere Portal offers many services and portlets for integrating with Lotus collaboration software. WebSphere Portal provides services for storing, protecting and retrieving user credentials in a *credential vault*. Single sign-on (SSO) is a service that WebSphere Portal offers for automatically authenticating users against various back-end systems through the implementation of disparate security models.

J2EE: Because WebSphere Portal is an enterprise application running on WebSphere Application Server, it can access services provided by a typical J2EE application server, which includes EJB components, Web services, JDBC data sources, JMS and servlets.

Agenda

- Portal, portlet and portal server defined
- WebSphere Portal Server overview
- ✓ **WebSphere Portal - the face of SOA**
- Portlet-application development

WebSphere Portal — the face of SOA

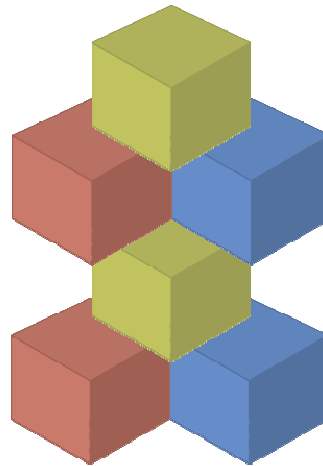
This next section of the course examines how WebSphere Portal acts as a front-end to SOA and how you can use it to promote an SOA.

What is a service-oriented architecture?

“An **approach** for **building** distributed systems that deliver application functionality as **services** to either user applications or other services”

Or more simply explained...

Composing business value out of IT assets



Services are the building blocks for reuse.

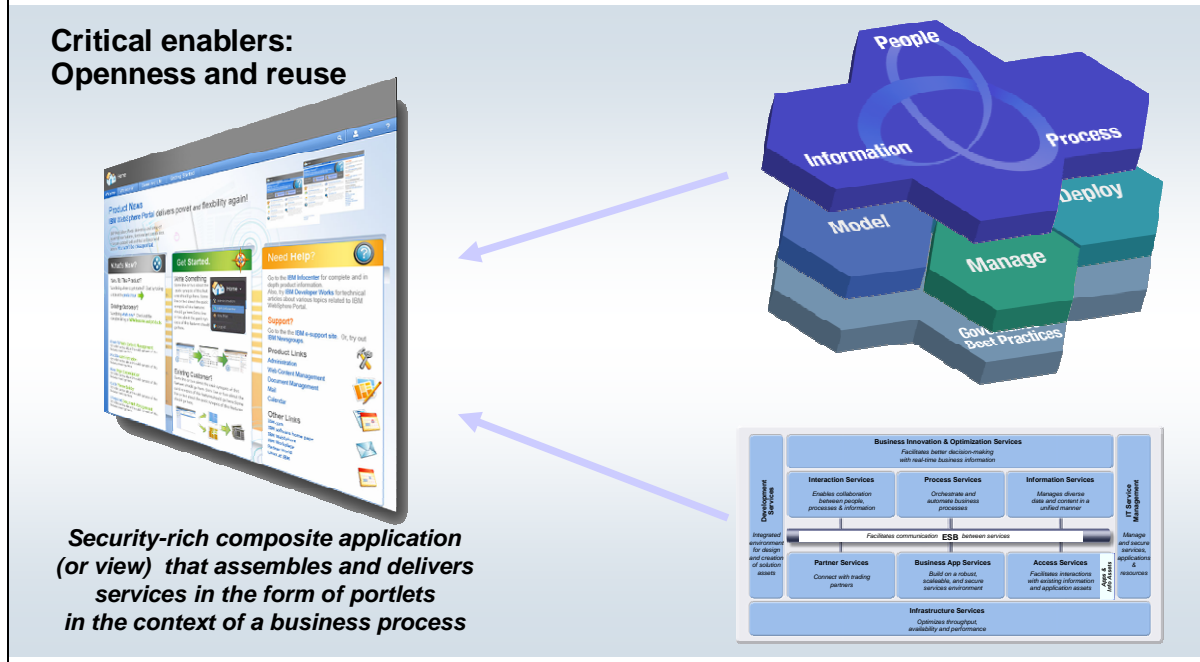
What is a service-oriented architecture?

SOA is an architecture that leverages open standards to represent software assets as services. It provides a standard way of representing and interacting with software assets. Individual software assets become building blocks that can be reused in developing other applications.

SOA shifts the focus to application assembly rather than implementation details and can be used internally to create new applications out of existing components. SOA can also be used externally to integrate with applications outside of the enterprise.

It is easy to understand the value of services if you think of them as building blocks for creating applications.

WebSphere Portal - front-end to SOA



WebSphere Portal — front end to SOA

You can assemble components into composite applications, which are a vital element of SOA. These composite applications are projected to users through a compelling front-end device — a portal.

The foundation tenet of SOA is the concept of *reuse*, creating business applications and processes that can be reused as needed, as with any other service. WebSphere Portal V6.0 delivers significant SOA enhancements through tools such as IBM WebSphere Portlet Factory, which allows business users (as opposed to J2EE developers) to develop and deploy composite applications and templates and, thus, allowing composite portal applications to be reused and replicated. Other SOA enhancements include support for application context, application roles, and the concept of membership and community.

With SOA, it is possible to leave the legacy application and reuse it in the portal environment where it is much easier to access than the original application. Major analysts say portals will be the first implementation of SOA for the majority of enterprises.

SOA and composite application capabilities enable businesses to be more flexible and responsive in how they deliver business solutions. SOA is not about a single product or solution; SOA is a mind set and architecture principle that looks at reusable business functions and business processes; and it assembles applications from these building blocks. This enables businesses to optimize what to *in-source*, what to *out-source* and what to *near-source*, bringing flexibility to the business model.

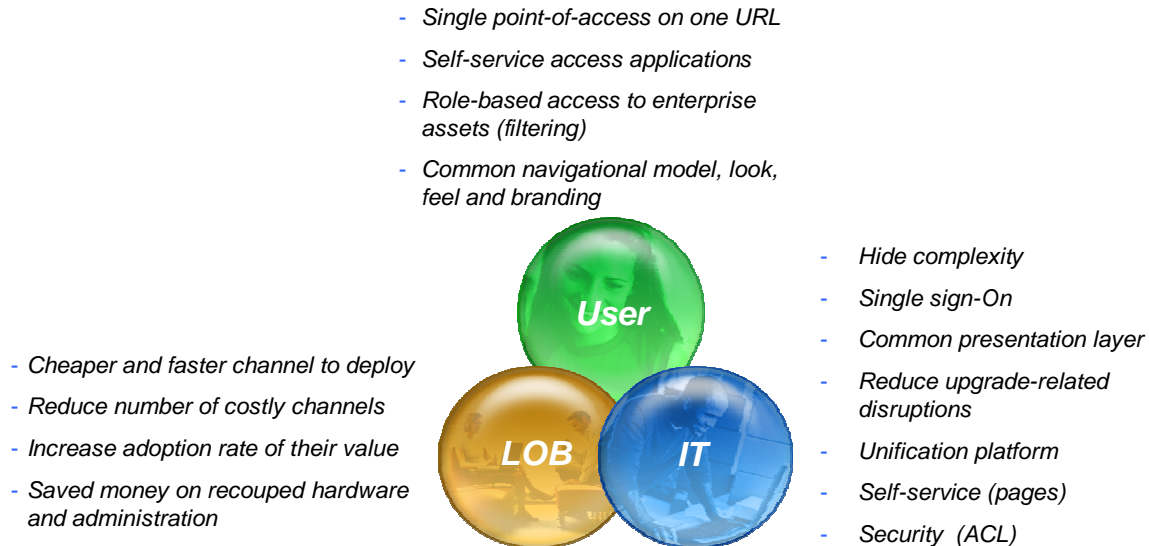
People often think about these things starting from the back-end, through integration, business process modeling, and optimizing the definition and use of services. A holistic approach extends

to, or starts from the front end, though. Then, people experience SOA in a very practical way, where users and services come closely together.

A portal is a composite application that is assembled at the front end. It is an aggregation point for services that are delivered through portlets. This new feature in WebSphere Portal V6.0 makes it easier for business users to access and customize portals to interact with services they need.

SOA and WebSphere Portal add value

Business value of portal for users, IT and line-of-business



SOA and WebSphere Portal add value

SOA and WebSphere Portal provide value for the business user, IT and the line of business.

For the *user*, these technologies present a single point-of access to many applications – all from one URL. Self-service access applications, SOA and the portal also provide role-based access to enterprise assets through filters, as well as a common navigational model, look, feel and branding.

For the *line of business*, SOA and WebSphere offer a cheaper and faster channel for deploying new function while also reducing the total number of costly channels. They accelerate the rate of adoption for the value they deliver and allow the enterprise to save money as a result of recouped hardware costs and reduced administrative burden.

For the enterprise's *IT department*, SOA and WebSphere Portal provide tremendous value, too, by hiding complexity and allowing support for a SSO process. These technologies deliver one common presentation layer by serving as a unification platform and, because of their nature, reduce upgrade-related disruptions. By nature, portals are highly effective in delivering self-service types of Web applications and the security of all portal-driven applications is ensured through access-control list (ACL)-based security.

The value of WebSphere Portal and SOA

1

Platform for composite applications

Reusable assets reduce engineering costs and deployment times

2

Platform for productivity

Dynamic, role-based composite views delivered in context

3

Platform for enterprise agility

Engineered to support open standards and deliver process-driven portals

The value of WebSphere Portal and SOA

Portals bring together vital information, applications and content. However, portals do more than that — they bring together processes and allow people to interact with them in interesting ways.

Processes are everywhere, whether formal workflows or the more ad hoc types of workflows that happen at the department level every day. A process takes place between portlets on the Web page. Simply put, a change occurs in one portlet that triggers an activity that is presented in another portlet. This intraportlet has many applications, including customer service and customer tracking, as well as order-status and supply-chain issues. Additionally, all of these processes are surfaced within a role-based environment that shows people only what they should see, based on their role and security level.

Agenda

- Portal, portlet and portal server defined
- WebSphere Portal Server overview
- WebSphere Portal - the face of SOA
- ✓ **Portlet-application development**

Portlet-application development

Now that basic portal and portlet concepts have been discussed, as well as the IBM offering in this category, WebSphere Portal, you can now begin to understand what is involved with portlet-application development. As you will see in the next section, portlet-application development is similar to Web-application development. However, portal and portlets are more feature-rich and robust.

Developing portals and portlets

- **Developers, Web designers and portal administrators can work independently**
- **Web designers can develop new themes and skins to change the look and feel**
- **Portal administrators put it all together by laying out portlets and pages**
- **An important activity is to create portlets**

Developing portals and portlets

Developers, Web designers and portal administrators can work independently when developing and changing the portal.

Web designers can develop new themes and skins to change the look and feel of the portal.

Portal administrators put it all together by laying out portlets and pages.

When developing portals and portlets, one of the main activities is to create portlets. Some portlets do not require any coding; however, there are several options for developing portlets from the ground up, as is explained next.

Portal and portlet development

Development option	Skills	Level	Level of integration with WebSphere Portal
Create custom portlets	Java programming	Advanced	Completely integrated
Use IBM tools to build portlets	Experience with IBM tools	Intermediate	Not as integrated as custom portlets
Use other builder tools	Experience with other vendors' tools	Intermediate	Depends on the tool
Buy portlet applications		Beginner	Depends on the application

Portal and portlet development

You can develop portals and portlets in any combination of methods. You can create them the ground up or use IBM tools to build them. You can also use other builder tools, or you can even buy portlet applications. This chart shows the level of skill required for each of these development options. All of these options can be used to integrate System i applications and data.

Agenda

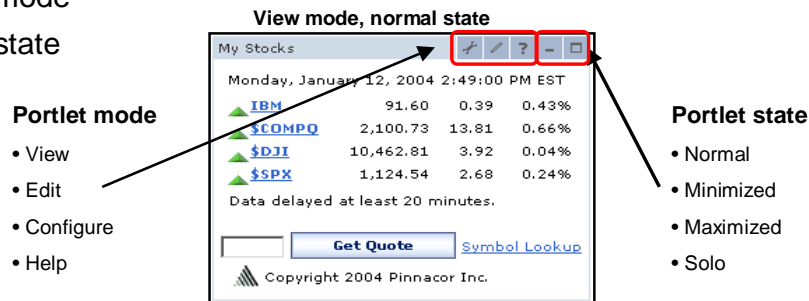
- Portal, portlet and portal server defined
- WebSphere Portal Server overview
- WebSphere Portal - the face of SOA
- Portlet-application development
 - ✓ Developing custom portlets
 - Using portlet builders from other vendors
 - Purchasing portlet applications
 - Making portlets from existing Web applications

Developing custom portlets

One of the more complex development options is to create portlets from the ground up. This method is the most complex because it requires a clear understanding of portal and portlet concepts and architecture. Although it is more complex, it also gives you the most power and allows for tighter integration into WebSphere Portal and the surrounding framework.

Portal applications

- **Portal applications are Web applications. However, there are some differences**
 - In a Web application, only one Web application is displayed in a browser at any given time.
 - In a portal, multiple applications share a browser.
- **Additional WebSphere Portal application concepts**
 - Portlet mode
 - Portlet state



Portal applications

In a portal, multiple applications share a browser. This is a key difference between applications that run in a portal and Web applications that are displayed, one-at-a-time, in a browser.

Portlet modes

An additional portlet concept allows a portlet to display a different user interface, depending on the task required of the portlet. The IBM Portlet API provides the following modes:

- **View:** When a portlet is initially constructed on the portal page for a user, it is displayed in its view mode. This is the portlet's normal mode of operation.
- **Help:** If the portlet supports this mode, the portlet provides a help page for users to obtain more information about the portlet.
- **Edit:** If the portlet supports this mode, the portlet provides a page for users to customize the portlet for their own needs. For example, a portlet can provide a page for users to specify their location for obtaining local weather and events. Users must be logged into the portal to access the edit mode.
- **Configure:** If the portlet supports this mode, the portlet provides a page for portal administrators to configure a portlet for all users of the portlet.

Portlets declare support for these modes in the portlet-deployment descriptor. All portlets must support the view mode. The IBM Portlet API provides the `PortletRequest.getMode()` and `PortletRequest.getPreviousMode()` methods for the portlet to determine the current or previous mode.

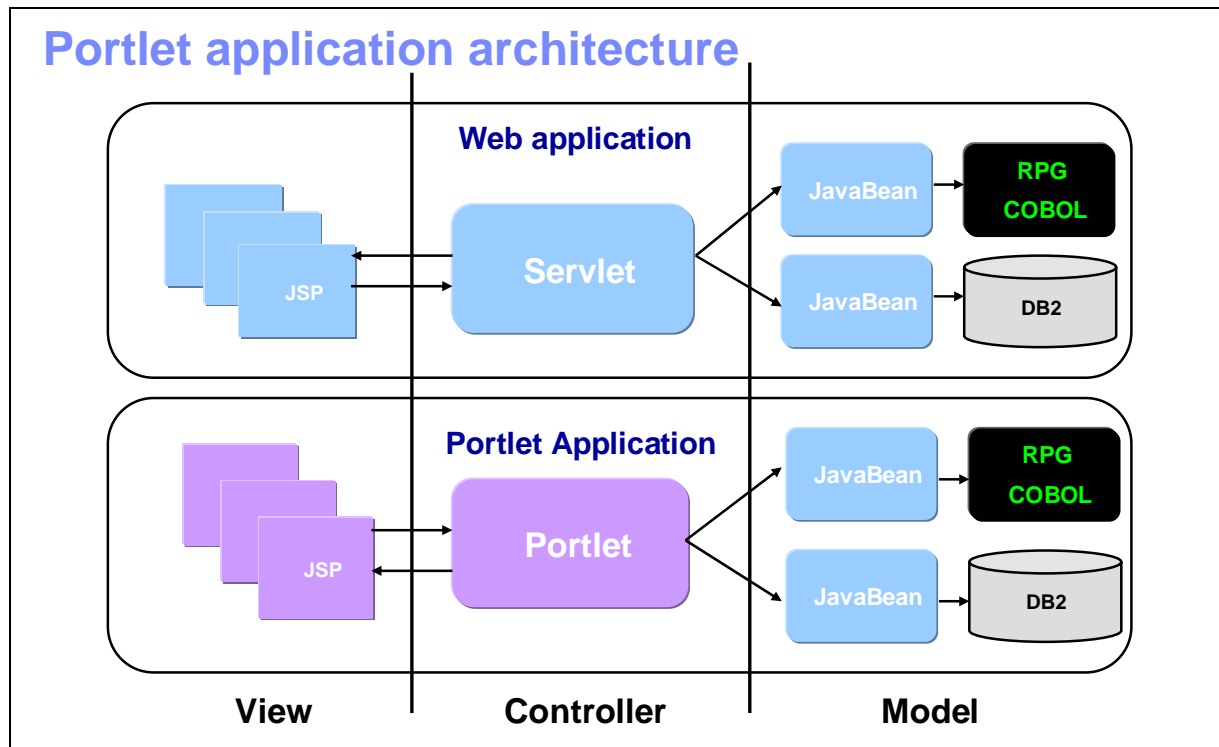
The portlet skin can provide links that allow a user to switch between portlet modes. Clicking one of these links changes the portlet's mode or state (see the following definition of *portlet state*). The links can be implemented in a context menu or they can be rendered within the title bar.

Portlet states

Portlet states allow users to change how the portlet window displays within the portal. Portlet states are maintained in the `PortletWindow.State` object with a Boolean value. The Portlet API provides the `PortletWindow.getWindowState()` method to determine the portlet's current state. The following states can be invoked by using icons in the title bar, in the same way that Windows applications are manipulated:

- **Maximized:** When a portlet is maximized, it is displayed in the entire body of the portal, replacing the view of other portlets.
- **Minimized:** When a portlet is minimized, only the portlet title bar is displayed on the portal page.
- **Normal:** When a portlet is initially constructed on the portal page, it is displayed in its normal state — arranged on the page along with other portlets. When the portlet is maximized or minimized, clicking a *restore* icon returns the portlet to its normal state.

Portlets declare support for these states in the portlet-deployment descriptor.

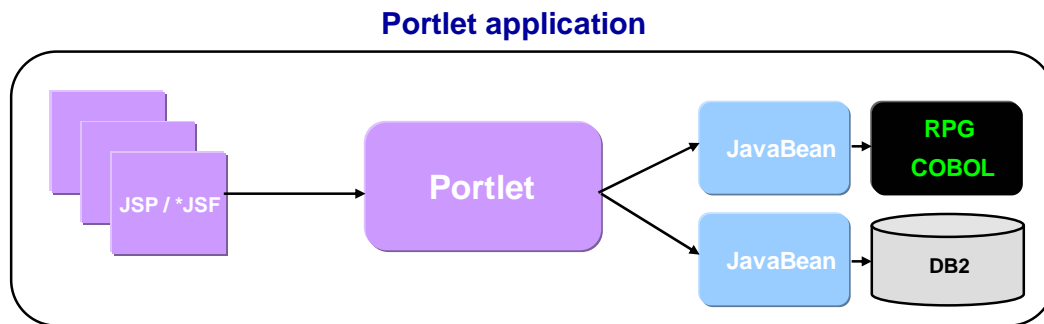


Portlet-application architecture

JSP pages are actually a type of servlet. Each JSP page is automatically converted into a servlet behind the scenes. Just as with a servlet, JSP pages provide developers with access to the *request* and *response* objects; you write your output to the response's output stream. Some developers make the mistake of completely replacing their servlets with JSP pages, but they are ideally used differently — in a manner that complements each other. JSP pages provide a convenient syntax for generating your output — a mixture of Java, special JSP syntax and browser markup (as with HTML and JavaScript) — whereas servlets, which are pure Java technology, provide a cleaner way to interact with your business data.

The *model-view-controller design*, which is familiar to many developers, makes it easier to use servlets and JSP pages together. The *model* part of the design is the business data and objects that help in reading and updating it. The *controller* part of the design is the servlet, which takes requests from the user's Web browser, acts on them using the model, and neatly packages the information that the JSP page needs into one object (usually called a *bean*). The JSP page is the *view* portion of the design that renders the information from the servlet in a clear and visually pleasing way. Any platform that supports servlets can also support JSPs.

Portlet application architecture



- JSP contains HTML and Java scriptlets and can also use portlet-tag libraries.
- One JSP is implemented for each mode.
- Portlet JSPs are JSP fragments and must not have HTML <body> tags.
- *Alternatively, use JSFs.
- Portlets are implemented with Java API (JSR 168 or the IBM Portlet API).
- Business logic can be implemented with various technologies.

Portlet application architecture (continued)

A JSP is a combination of HTML and Java scriptlets. JSPs also use portlet-tag libraries. A single JSP is required for each mode that you implement. Portlet JSPs are JSP fragments; they must not have HTML <body> tags. Use JSF as an alternative.

Portlets are implemented with the Java API: JSR 168 or the IBM Portlet API

You can use various technologies, either manually or with visual development tools, to implement business logic.

Note: The use of JavaScript is not recommended in portlets, because it can lead to unpredictable results.

Portlet JSP example

```
<%@ page session="false" contentType="text/html"
import="java.util.*,javax.portlet.*,com.ibm.flight400.portlets.*, com.ibm.flight400.jdbc.*" %>
<%@taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<portlet:defineObjects/>

<%
DBPortletSessionBean sessionBean =
(DBPortletSessionBean)renderRequest.getPortletSession().getAttribute(DBPortlet.SESSION_BEAN);
%>
<FORM method="POST" action="<portlet:actionURL/>"
...
</table>
...
<% // Display search results
Vector searchResults = sessionBean.getSearchResults();
if(searchResults !=null){
Iterator iter = searchResults.iterator();
FlightInfo flight = null;
while(iter.hasNext()){
flight = (FlightInfo)iter.next();
%>
<tr bgcolor="#bfe6ee">
<td><%=flight.getDepartCity()%></td>
<td><%=flight.getDepartTime()%></td>
</tr>

<% // Close if and for loop
}} %>
</table>
</FORM>
```

Tag library

Tag

Java scriptlet

Portlet JSP example

Glance at this sample portlet JSP shown here. Notice its use of the tag library. You can also see the tag itself, as well as the Java scriptlet that is embedded in the portlet.

Portlet API: IBM Portlet API or JSR 168

- **Although IBM Portlet API and JSR 168 are different in their implementations, the functionality that they provide is similar.**
- **In this presentation, we focus on concepts that apply to both APIs.**
- **WebSphere Portal Server supports portlets written in IBM Portlet API and JSR 168.**
- **Deciding which API to use**
 - Start with JSR 168
 - Use IBM Portlet API only if you need additional features that are not available in JSR 168

Portlet API: IBM Portlet API or JSR 168

Although IBM Portlet API and JSR 168 are different in their implementations, the functionality that they provide is similar. For example, both APIs support the following functions: request processing, customization, messages and events and multiple devices.

All references in this presentation focus on concepts that apply to both APIs.

WebSphere Portal Server supports portlets written in IBM Portlet API and JSR 168. Therefore, you might be confused when trying to decide which of these APIs to use. The recommendation is that you start with JSR 168; use IBM Portlet API only if you need additional features that are not available in JSR 168.

An excellent IBM Portlet API and JSR 168 comparison white paper is available at:
ibm.com/developerworks/websphere/library/techarticles/0406_hepper/0406_hepper.html

A link to this Web site is available in the links section of this course.

Why use the JSR standard?

- **Portlets are platform-agnostic**
 - Can be deployed to any portal product that conforms to the specification
- **IT managers**
 - Can support multiple portal products to accommodating various business needs
 - Can deploy to all compliant portal frameworks without much change
- **Developers**
 - Can reuse code – create and maintain one set of JSR 168 compliant portlets
 - Run on any compliant portal server with few, if any, modifications.
 - Benefit from extra functionality beyond the standard servlet functionality (because this API is specifically designed for portlet creation)

Why use the JSR standard?

The primary reason for using the JSR standard portlet API is this: portlets that adhere to this specification are portal-product-agnostic — which, after all, is one of the major attractions to using Java in the first place. You can deploy JSR 168 portlets to any portal product that conforms to the specification.

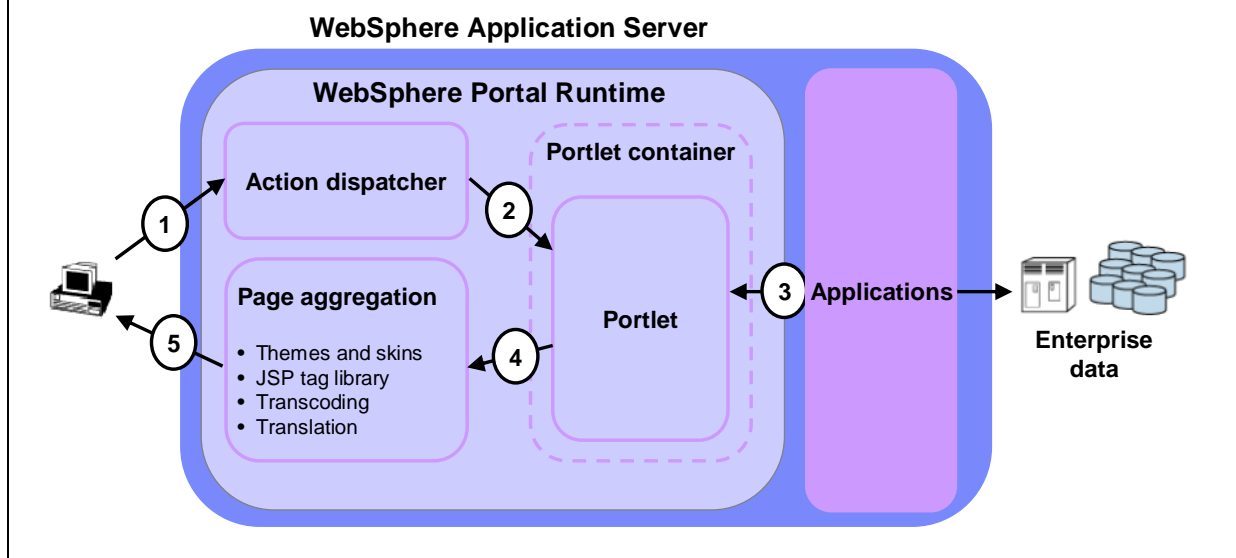
For *IT managers*, using the JSR standard gives them the ability to support multiple portal products, thus accommodating the unique business needs of various departments and audiences. That is, compliant portlets can be deployed to all compliant portal frameworks without extensive engineering changes.

For *developers*, using the JSR standard allows them to reuse code. They can create and maintain one set of JSR 168-compliant portlets. Portlets can run on any JSR 168 Portlet Specification-compliant portal server with few, if any, modifications. Because this API is designed precisely for portlet creation, developers benefit from the extra functionality it offers – beyond the standard servlet functionality.

For more detail, see the Sun Microsystems white paper *Introduction to JSR 168 – The Java Portlet Specification* at http://developers.sun.com/prodtech/portalserver/reference/techart/jsr168/pb_whitepaper.pdf

A link to this Web site is available in the Links section of this course.

Portlet API: Request processing



Portlet API: Request processing

This chart shows the flow of processing for a portlet request:

1. The user clicks the portlet link on the portal Web site. This causes a request to be sent to the action-dispatcher function.
2. The action dispatcher sends input to the portlet as a request object.
3. The portlet accesses the application (which invokes business logic) that accesses data and returns the data to the portlet.
4. The portlet generates markup fragments and passes the marked-up data to the page-aggregation function.
5. The page-aggregation function builds the HTML page and returns it to the browser (that is, it refreshes portal page so that the user can view the results of the request).

Portlet API: Customization

■ Customization APIs

- Customize information for a group of users
For example: Management team can only view sensitive data
- Customize information for individual users – administrator-controlled
For example: Allow users to modify their stock ticker

■ Multiple instances of a portlet on the same page

- Use customization APIs to display different information
- Users can customize portlets by going to the Edit mode

The diagram shows a 'World Clock' portlet interface. On the left, there are two purple boxes: 'Group settings' and 'Individual settings'. Arrows point from these boxes to the portlet's output. The 'Group settings' box points to the 'Local Time' section, which displays '11:07 AM Eastern Time (US & Canada) - 11/12/03'. The 'Individual settings' box points to the 'Time Zone' table, which lists 'Alaska' and 'Amsterdam, Copenhagen' with their respective local times. Below the table is a 'Quick Search' input field.

Time Zone	Local Time
Alaska	7:07 AM (11/12/03)
Amsterdam, Copenhagen	5:07 PM (11/12/03)

Portlet API: Customization

You can also customize portlets on a user-group basis. You can assign permissions to some user groups to use particular portlets; those without permissions to use the portlets do not even see them in their portal.

A portlet can also change its output based on the user groups with which the current user is associated; this is possible because the portal server provides an easy way to determine group membership. The portal server can use an existing LDAP directory to import user groups, or you can create a new user directory.

Yet another type of portlet customization occurs because of user preferences. These customizations are subject to the limits that the portal administrator sets, but within these limits, users can decide which portlets are in their portals, what pages those portlets are on and any other preferences you choose to let them change. For example, you can allow users to customize their stock-ticker portlet with their own portfolio of stocks and store that information in the PortletData object for that user. When the user logs out and returns later, the saved preferences are preserved from the previous session.

Portlet API – messages

- **ActionListener:** Turns user clicks in your pages into action events, an easy-to-use model familiar to many Java programmers
- **WindowListener:** Notifies your portlet of window events from the portal – such as when your portlet is maximized, minimized and restored
- **MessageListener*:** Lets you handle messages sent from one portlet to another.

* IBM Portlet API only

Portlet API — messages

Another important feature provided by the Portlet API is the ability to handle messages and events. The ActionListener interface turns user clicks in your pages into action events; this is an easy-to-use model that is familiar to many Java programmers. The WindowListener interface notifies your portlet of window events from the portal, such as when the user has maximized, minimized or restored a portlet. The MessageListener interface lets you handle messages sent from one portlet to another. However, it is important to note that this interface is only available with the IBM Portlet API.

Portlet API – events

- **ActionEvents:** Fires when portal receives an HTTP request that is associated with an action (for example, the user clicks a link or button)
- **WindowEvents:** Fires when the user changes the state of the portlet window
- **MessageEvents*:** Fires when a portlet sends a message to another portlet

* IBM Portlet API only

Portlet API — events

Portlet events contain information about an event that might trigger the need for a portlet to respond. For example, when a user clicks a link or button, this generates an action event. To receive notification of the event, the portlet must have an event listener implemented within the portlet class.

Action events are generated when an HTTP request is received by the portlet container that is associated with an action, such as when the user clicks a link. To receive action events, the portlet class must implement the `ActionListener` interface and a portlet action. A portlet action can be one of the following types: a simple portlet-action string or a `portletAction` object.

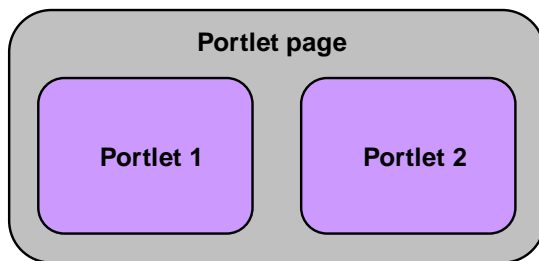
With the simple portlet-action string, actions that are created as simple actions can be run multiple times, enabling a user's back button to work. Links created with simple portlet actions are represented in the URL rather than in the session. Therefore, portlets with simple actions can be placed on an anonymous page where no session exists.

The `PortletAction` object has been deprecated in favor of simple portlet action strings. It is maintained in the Portlet API to support existing portlets that use `PortletActions`. You always need to use simple portlet actions instead of `PortletAction` objects.

Message events are generated when another portlet within the portlet application sends a message. This function is only available with the IBM Portlet API and is discussed in greater detail shortly.

Portlet API – two-phase processing

- **Two-phase request processing in a portlet involves the following two types of processing:**
 - Action processing
 - Request processing
- **Action processing is guaranteed to complete before a portlet is called to render.**
- **Portlets should use the service phase only to render portlet output.**



1. An action event is fired from Portlet 1.
2. `actionPerformed()` is called in Portlet 1.
3. `service()` is called in Portlet 1.
4. `service()` is called in Portlet 2.

Portlet API — two-phase processing

A portlet has a different processing and rendering sequence than a servlet. A servlet does all of its processing in the `service()` method. A portlet, on the other hand, uses two-phase processing that is split between an event phase and a render phase. This split is necessary to accommodate communication between portlets before rendering output in the service stage. The event phase is guaranteed to complete before a portlet is called to render.

During the event phase, the portlet implements an `ActionListener` interface, which provides the `actionPerformed()` method, to which an `ActionEvent` object is passed. When a user clicks a link or a **submit** button, an `ActionEvent` can be generated. The portlet action can be obtained from the `ActionEvent`, which describes the triggering event. When the `actionPerformed()` method is invoked, a response object is not available because this is not a rendering step. All state changes must be handled during action processing.

Portlets should use the render phase only to render portlet output; do not place logic in the rendering phase. The `service()` method is not only called following the `actionPerformed()` processing when a user clicks on a link or button in a portlet, but is also called when the portal page refreshes. Thus, given a page with two portlets, A and B, when the user clicks a link in portlet A, the `actionPerformed()` and `doView()` methods are called for portlet A, but only the `doView()` method is called for portlet B. After the content-generation phase has started, no further events are delivered. For example, messages cannot be sent from within the `beginPage()`, `service()` and `endPage()` methods. The resulting message event is not delivered; it is essentially discarded.

The event listener is implemented directly in the portlet class. The listener accesses the `PortletRequest` from the event and responds using the `PortletRequest` or `PortletSession` attributes.

Note: AJAX allows for re-rendering of a single portlet, instead of all portlets.

IBM Portlet API – messaging

- **Single-addressed messages:** Messages sent to a specific portlet by specifying the portlet name on the send() method.
- **Broadcast messages:** Messages sent to all portlets on the page.
- **Can send messages to portlets located on the same page (same or different portlet applications)**
- **Cannot send messages to portlets on other pages**

IBM Portlet API — messaging

IBM portlets that either implement the PortletMessage interface or extend the DefaultPortletMessage class can send information to other portlets on the page. Portlets receiving the message must implement the MessageListener interface and an object with the type PortletMessage. However, it is recommended that you use the property-broker service for interportlet communication.

Message events can be sent from one portlet to others if the recipient portlets are members of the same portlet application and are placed on the same page as the sending portlet. Additionally, a DefaultPortletMessage can cross portlet-application boundaries and might be sent to all portlets on a page. A MessageEvent can be sent actively to other portlets only when the portlet is in the event-processing cycle of the portlet container, otherwise an exception is thrown. There are two types of messages:

- **Single-addressed messages** are messages sent to a specific portlet by specifying the portlet name on the send() method.
- **Broadcast messages** are messages sent to all portlets on the page.

Message events are useful when changes in one portlet must be reflected in another one. An object with the type PortletMessage has to be implemented, which is passed by means of the MessageEvent. The portlet that receives the message must implement the MessageListener interface and an object with the type PortletMessage.

IBM Portlet API – click-to-action (C2A)

- **A communication mechanism for independent portlets**
- **Can broadcast properties to multiple portlets**
- **A “wire” can be created to send messages without user action**

IBM Portlet API — click-to-action

Click-to-action (C2A) provides a platform for portlet messaging and an intuitive GUI for users. With a click of the mouse, C2A facilitates easy data transfer from a source portlet to the target portlet. When the user clicks the C2A-enabled icon in the source portlet, a pop-up menu is displayed that contains a list of possible targets for the action. After the user selects the target, the built WebSphere Portal property broker delivers the data to the target in the form of a corresponding portlet action, causing the target portlet to react and display a new view with the results.

IBM Portlet API messaging and C2A

Messaging	C2A
Portlets are tightly coupled	Portlets are loosely coupled
Can send any Java object that implements PortletMessage	Can send a single string value that is visible to a user
Programmer-controlled integration of portlets	User-controlled integration of portlets
Design-time integration	Runtime integration

IBM Portlet API messaging and C2A

This chart illustrates the contrast in characteristics of using messaging with the IBM Portlet API as compared to using C2A with the IBM Portlet API for the same purposes.

With messaging, portlets become tightly coupled, whereas C2A allows the portlets to be loosely coupled.

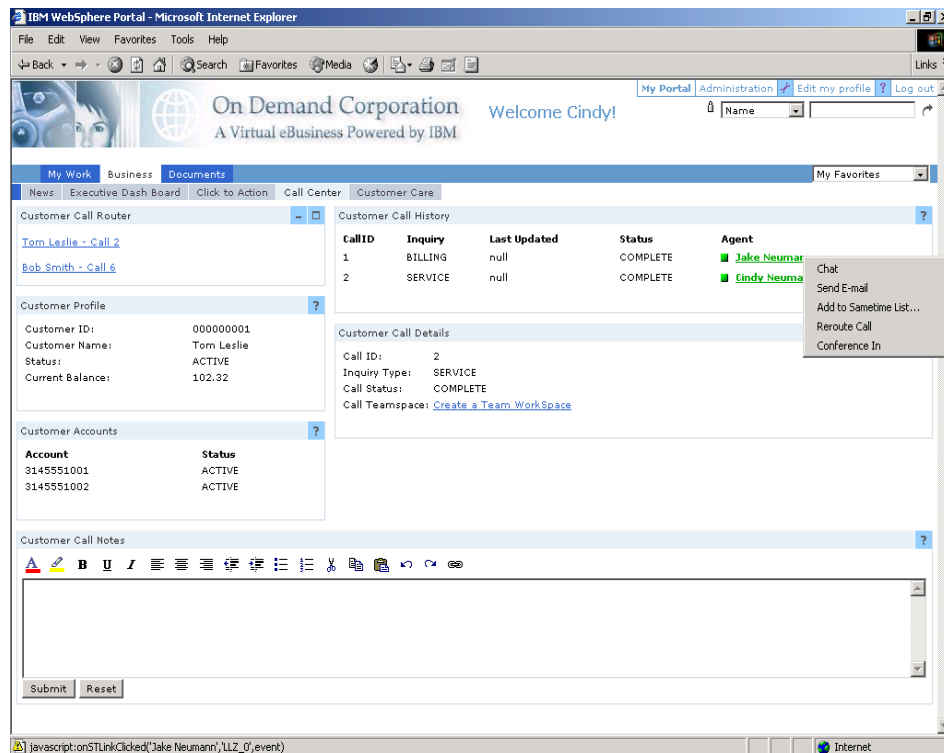
With Messaging, it is possible to send any Java object that implements PortletMessage. In contrast, C2A supports the sending of a single string value that is visible to the user.

With messaging, the developer must control the level of portlet integration. C2A allows the user to control the level of portlet integration.

Messaging forces design-time integration efforts; C2A supports runtime integration of portlets.

A sample application – portlets in action

- Call-router populates the call-center employee's desktop
- People awareness allows two-way chats
- People tag can reroute the call.



A sample application — portlets in action

In this sample portlet application, information from a call-router portlet can populate the call-center employee's desktop so that he or she can more rapidly assist the customer who has called in for assistance.

A people-awareness portlet is also available on this portal to allow the customer to chat with customer-service representatives. The employee can see that this method of support has successfully helped this same customer in the past.

A people-tag portlet can be extended with application-specific capabilities, such as rerouting the call. This can be of value if the call needs to pass to a second level of customer-support expertise to resolve the issue.

Portlet API – multiple-device support

- Built-in support to automatically use different JSP pages for different user devices
 - For example, browsers, PDAs or other mobile devices
- Portal server chooses the appropriate interface for each device
- Support for multiple portlet controllers allows different page and action sequences for each device type
- No need to program to a lowest common denominator; provide multiple interfaces for different device types, reusing the same underlying business logic

Portlet API — multiple-device support

Portlets also have many standard features that are not available to servlets at all. One key feature is built-in support to automatically use different JSP pages for various user devices. This lets you write portlets that work on many devices, including desktop computers with modern Web browsers, older or palmtop computers with limited Web browsers, PDAs and Web-enabled wireless phones. You do not need to program to the lowest common denominator (technologically speaking); you can provide multiple interfaces for different device types, reusing the same underlying business logic; the portal server chooses the most appropriate one for each user. You can even have multiple portlet controllers, which allows you to have different page and action sequences for each device type.

Portlet API – themes

- **Another key feature – the ability to customize look and feel**

- Theme development tools
- Copy and modify an existing theme
 - JSP pages (default, banner, navigation)
 - Images
 - Stylesheets
- Deploy the new theme in WebSphere Portal Server

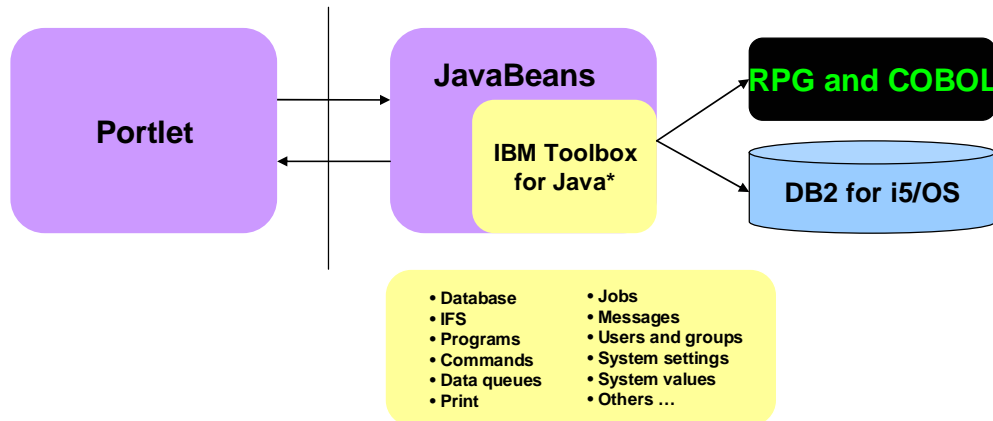


Portlet API — themes

Yet another key feature of portlets is the ability to customize them in a number of ways. When developers use the standard cascading style sheet that the portal server provides, all of their portlets have the same look and feel. Companies can create themes and skins for their portlets that reflect their individual style, and all the portlets (even those developed by a third party) change to that new style automatically.

Integrating with System i applications and data

- **IBM Toolbox for Java and JTOpen:** A library of Java classes and APIs for accessing System i programs, functions and data



Integrating with System i applications and data

The IBM Toolbox for Java, which is available as a cost-free download from the Web site below, contains a library of very useful Java classes and APIs for accessing System i programs, functions and data: ibm.com/servers/eserver/iseries/toolbox

Note: JTOpen is the open version of IBM Toolbox for Java and is downloadable from the Web site listed here.

There is a link to this Web site in the Links section of this course.

Integrating with System i

- **Common integration scenarios**

- Program call
- Database integration
- Web services

- **All integration options are APIs from IBM Toolbox for Java**

- Shipped with every System i model: JT400.jar
- Shipped with WebSphere Development Studio Client for System i
- Open source version: JTOpen.jar

Integrating with System i

When you plan for the integration of applications on the System i platform, you will certainly become familiar with IBM Toolbox for Java. Common integration scenarios include the use of a program call to initiate stored procedures or other RPG code. There are also some tools for connecting to and taking advantage of the sophisticated DB2 database that is provided with the delivery of every System i model ever since the family's inception a couple of decades ago. Additionally, the toolbox has tools for building Web services that encompass RPG, COBOL and DB2 resources.

All integration options are provided as APIs from IBM Toolbox for Java. This toolbox is conveniently shipped with every System i model, contained in a Java Archive (jar) file called JT400.jar. These tools are also shipped with the WebSphere Development Studio Client for System i product.

Note: The downloadable, open-source version of JT400.jar is JTOpen.jar.

Agenda

- Portal, portlet and portal server defined
- WebSphere Portal Server overview
- WebSphere Portal - the face of SOA
- Portlet-application development
 - Developing custom portlets
 - ✓ Using portlet builders from other vendors
 - ✓ Purchasing portlet applications
 - Making portlets from existing Web applications

Other portlet builders and purchased portlet applications

From a portlet-application development standpoint, yet another available implementation option is to use any of the many portlet builders from IBM Business Partners, or you can purchase portlet applications as well.

WebSphere Portal Business Solutions Catalog

- 1500+ prebuilt portlets*
- 270 Business Partners* provide portlets, solutions, services and other tools
- 26 Business Partners* remarket WebSphere Portal
- 1200 Business Partners* have WebSphere Portal training
- * Gain valuable exposure
- * Three months of free voice and e-mail support

* As of April 2007



WebSphere Portal Business Solutions Catalog

Be recognized for your innovation. Test your products against pre-established criteria to ensure compatibility with certain IBM offerings. Receive development and integration assistance during the validation process. Gain valuable exposure to IBM Business Partners from a variety of IBM PartnerWorld® offerings and Web sites.

Upon successful validation, earn the right to display specially designed IBM emblems, logos and marks in the packaging and marketing materials of your qualified offerings. These licensed trademark materials can help you identify and promote your software, hardware, services or technology solution as a qualified offering that meets compatibility and integration specifications established by IBM.

IBM Business Partners who want to list their system integration or product integration solutions in the *IBM Workplace Solutions Catalog* and *IBM Global Solutions Directory* to help differentiate themselves from the competition and gain exposure to IBM clients and sales teams.

Business Partners who want to earn the right to use the “Ready for IBM WebSphere Portal” or the “Ready for IBM Lotus Domino” software mark can help show their customers their commitment to quality.

The *IBM Workplace Solutions Catalog*, which features industry-specific and horizontal solutions for on demand business, gives you targeted exposure directly to potential and current users of the IBM Workplace™ family of products, including Lotus Domino and WebSphere Portal software. The *Global Solutions Directory* gives you worldwide exposure to IBM clients.

To find more details, go to <http://catalog.lotus.com/wps/portal/portal>

There is a link to this Web site in the Links section of this course.

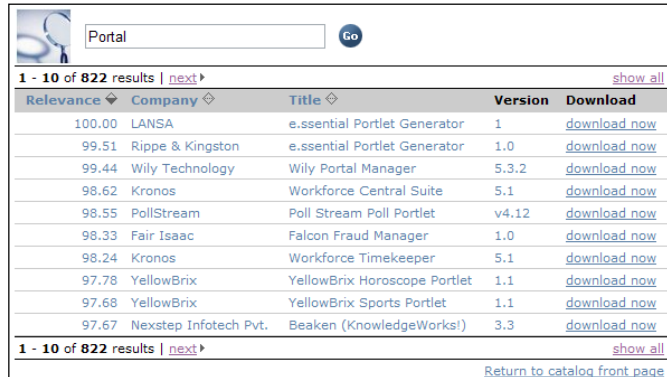
Portlet-builder tools

■ Portlet builders

- From Business Partners and IBM
- Fee, free, cost-free trial

■ Search WebSphere Portal Catalog for portlet-builder solutions:

- IBM Web-page portlet
- WebSphere Portal Theme Builder
- Lotus Domino Application Portlet (DAP)
- IBM Lotus Notes® and Domino as well as Extended Products Portlets
- WebSphere Portal sample portlets
- **Many, many more**



The screenshot shows a search interface with a search bar containing the word 'Portal' and a 'Go' button. Below the search bar, it indicates '1 - 10 of 822 results' with a 'next' link. A table lists the search results with columns for Relevance, Company, Title, Version, and Download. The table contains 10 rows of results, each with a relevance score, company name, product title, version number, and a 'download now' link. At the bottom of the table, it again shows '1 - 10 of 822 results' with a 'next' link and a 'Return to catalog front page' link.

Relevance	Company	Title	Version	Download
100.00	LANSA	e.sential Portlet Generator	1	download now
99.51	Rippe & Kingston	e.sential Portlet Generator	1.0	download now
99.44	Wily Technology	Wily Portal Manager	5.3.2	download now
98.62	Kronos	Workforce Central Suite	5.1	download now
98.55	PollStream	Poll Stream Poll Portlet	v4.12	download now
98.33	Fair Isaac	Falcon Fraud Manager	1.0	download now
98.24	Kronos	Workforce Timekeeper	5.1	download now
97.78	YellowBrix	YellowBrix Horoscope Portlet	1.1	download now
97.68	YellowBrix	YellowBrix Sports Portlet	1.1	download now
97.67	Nexstep Infotech Pvt.	Beaken (KnowledgeWorks!)	3.3	download now

Portlet-builder tools

Portlet-builder tools are available from both IBM and its Business Partners. Some of these tools are cost-free; others are available for a fee, but they usually offer a free-trial period for you to determine if it suits your development needs.

To find these providers, go to the IBM catalog site and search the *WebSphere Portal Catalog* for portlet-builder solutions. Some of the tools and solutions include the offerings you see on this chart. However, there are many, many more.

Look for a companion course to this one, entitled “Understanding IBM portal tools.” It provides more detail on the portal- and portlet-development tools that help you build enterprise-class portal sites.

Agenda

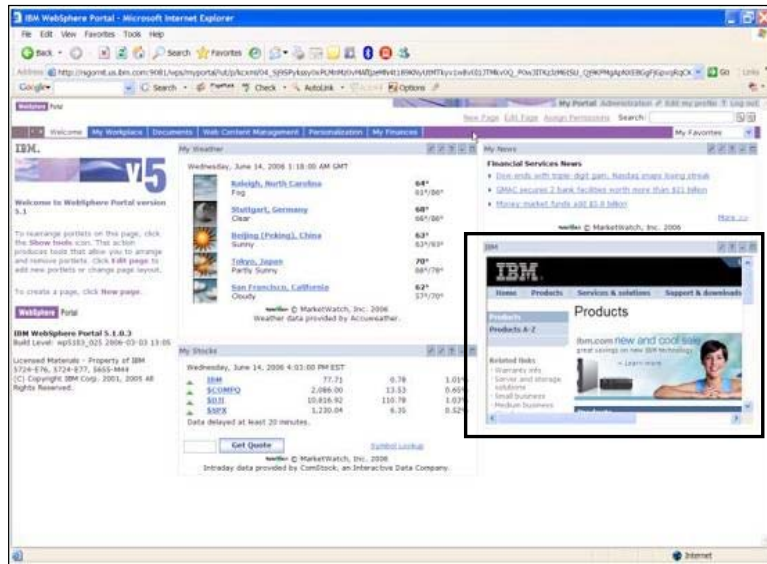
- Portal, portlet and portal server defined
- WebSphere Portal Server overview
- WebSphere Portal - the face of SOA
- Portlet-application development
 - Developing custom portlets
 - Using portlet builders from other vendors
 - Purchasing portlet applications
 - ✓ **Making portlets from existing Web applications**

Making portlets from existing Web applications

And finally, the next section of this course discusses how you can migrate a Web application into a portlet.

Web applications and portal applications

- **Web applications need a container to run in Portal**
- **Possible containers**
 - iFrame portlet or Web-page portlet
 - Web-clipping portlet
- ...or develop from the ground up



Web applications and portal applications

For Web applications to run in a portal, you must package them in a container. A few examples of containers that are becoming popular among developers include the iFrame portlet (also called a Web-page portlet) and a Web-clipping portlet.

iFrame and Web-page portlets display a Web application in WebSphere Portal. However, there are some limitations. For instance, iFrame and Web-page portlets do not use the IBM Portlet API and, hence, do not support features of WebSphere Portal such as single sign-on, messaging, customization and others. Additionally, the application state is lost when a portlet is resized or a page is refreshed (for example, when another portlet is modified) — this can be quite tricky to attempt to work around.

In contrast to iFrame and Web-page portlets, a **Web-clipping portlet** has the mission of displaying content from other Web sites. This solution is useful for displaying static content or a small part of the dynamic content that is presented on another Web site. To implement Web-clipping, you must invoke transcoding technology to convert the Web pages so that they comply with the Portlet API. The limitation to using Web-clipping portlets is the same as for iFrame and Web-page portlets — there is no support for WebSphere Portal features. Additionally, not all Web-site implementations are capable of being clipped. There might also be some unpredictable behavior when a Web site that is being clipped needs to maintain state.

Portalizing from the ground up

- **Portalizing a Web application: migrating the Web Application to use Portlet API and run natively in WebSphere Portal Server**
- **Main tasks:**
 - Evaluate presentation layer: a Web application can be converted to multiple portlets that can interact with each other
 - Change JSPs to comply with Portlet API
 - Implement communication mechanism between portlets: messaging and click-to-action
 - Replace the controller layer (servlets) with portlets

Portalizing from the ground up

Of course, you can always develop your own container that migrates the Web application to use the Portlet API and run natively in WebSphere Portal Server. This might be necessary if you need a means of preparing applications for a portal that more narrowly fits your particular situation. If you decide to do this, there are four main tasks, as follows:

1. Evaluate the presentation layer. (Remember, you can convert a Web application to multiple portlets that interact with each other.)
2. Change the JSPs to comply with the Portlet API.
3. Implement a communication mechanism between portlets (for example, messaging and C2A).
4. Replace the controller layer (servlets) with portlets.

Another consideration when moving an application onto a portal site is the fact that some Web applications take over the entire browser window, with no opportunity to scale down the presentation.

Portal application deployment

1. Export a portal application as a WAR file
2. Deploy the portal application in WebSphere Portal Server
3. Set permissions on the portlets
4. Add the portlets to the Portal pages

Portal-application deployment

After you have packaged your Web application in a portal container, there is a four-step process to finally present the application to users over the portal, as shown on this chart.

Summary

- **Defined a portal, portlet and portal server**
- **Gave product overview of the IBM portal offering**
- **Briefly discussed how Portal acts as the front-end to an SOA**
 - **Reuse:** Reusable assets reduce engineering costs and deployment times
 - **Productivity:** promoted by dynamic, role-based content
 - **Openness:** standards-based to allow for enterprise agility
- **Discussed portlet application development options including:**
 - Developing custom portlets and the IBM Toolbox for Java
 - Other-vendor portlet builders
 - Purchase portlet applications via the WebSphere Business Solutions Catalog
 - Portalizing existing Web applications (IFrame, Web-clipping, from scratch)

Summary

This course covered a lot of material in the past hour. You should now understand the meanings and value of a portal, a portlet and a portal server. There was also an explanation of the IBM portal offering, WebSphere Portal Server. You know that the portal acts as the front end to a service-oriented architecture (SOA). The important thing to remember about SOA is that it provides a flexible means of reusing application assets, which helps reduce engineering costs and deployment timeframes. Because of its theme of reuse, SOA increases developer productivity. And because SOA is based on an open standard, it supports greater flexibility for the enterprise as new needs arise that require the support of IT resources and infrastructure.

In this course, you became aware of several development options for building and incorporating portlet applications. For instance, you can develop custom portlets and use IBM Toolbox for Java (or JTOpen) for its wealth of APIs and tools that allow you to have access to System i resources, including its award-winning, integrated DB2 database. You are now more familiar with the fact that a wealth of portlet builders and tools is available from IBM Business Partners — literally thousands of Business Partners post their software offerings on the IBM PartnerWorld solutions and catalog Web sites.

Finally, the presentation showed what is involved in portalizing your existing Web applications. You can do this by using iFrame or a Web-page portlet, as well as using a Web-clipping portlet. Depending on the Java skills the enterprise has in-house, you can modify your current Web application code from scratch.

One further reminder, be sure to look for a companion course to this one, entitled “IBM portal tools.” It provides more detail on the portal- and portlet-development tools that can help you build an enterprise-class portal site.

Resources

- IBM eServer i Information Center
<http://publib.boulder.ibm.com/series>
- IBM Publications Center
www.elink.ibm.link.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US
- IBM Redbooks™
www.redbooks.ibm.com
 - *WebSphere Portal v6 Best Practices* (SG24-7387)
 - *Best Practices for migrating to WebSphere Portal V6* (redp4227)
 - *Best Practices for WebSphere Portal* (redp4100)
 - Other programming guides, best practices, guidelines, handbooks and product information
- WebSphere Portal home page
ibm.com/websphere/portalfamily
- Portal Zone for developers
ibm.com/websphere/developer/zones/portal
- WebSphere 5.1 Portal demonstration site
www.wps51.dfw.ibm.com/wps/portal
- WebSphere 6.0 Portal demonstration site (requires cost-free registration)
<http://portal.ibmdemo.com/wps/portal>
- WebSphere Portal offerings and their features
ibm.com/software/genservers/portal/features
- IBM Systems Workload Estimator
www-912.ibm.com/wle/EstimatorServlet
- *Portlet API comparison white paper: JSR 168 Java Portlet Specification compared to the IBM Portlet API*
ibm.com/developerworks/websphere/library/techarticles/0406_hepper/0406_hepper.html
- IBM Toolbox for Java and JOpen download Web site (also contains a library of Java classes and APIs)
ibm.com/servers/eserver/series/toolbox
- *IBM WebSphere Portal Tuning Guide*
ibm.com/servers/enable/site/education/abstracts/wsptuning_abs.html
- IBM Systems Workload Estimator
www-912.ibm.com/wle/EstimatorServlet

- EMEA Techline
<http://w3.ibm.com/support/emea/techline/sizing.html>
- Americas Techline:
<http://w3.ibm.com/support/americas/techline/sizewise.html>
- Business Partner Techline
ibm.com/partnerworld/techline
- PartnerWorld Virtual Innovation Center™
ibm.com/isv/welcome/vic.html
- IBM System i Benchmark Center
ibm.com/servers/eServer/iSeries/benchmark/CBC/index.html
- Sun Microsystems white paper *JSR 168 – The Java Portlet Specification*
developers.sun.com/prodtech/portalserver/reference/techart/jsr168/pb_whitepaper.pdf

Trademarks and special notices

© Copyright. IBM Corporation 1994-2007. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

AIX, CloudScape, DB2, Domino, i5/OS, IBM, the IBM logo, IBM Virtual Innovation Center, iSeries, Lotus, Notes, OS/400, PartnerWorld, POWER5, Redbooks, System i, WebSphere, Workplace, workplace Forms and xSeries 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, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Information is provided "AS IS" without warranty of any kind.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.