



IBM Software Group

# IBM Rational Application Developer V6

## *Portal Tools*



@business on demand.

© 2004 IBM Corporation  
Converted to video July 7, 2015

This presentation will focus on the IBM Rational Application Developer V6.0 Portal Tools.

## Goals

- Introduce Portal Development Tools functionality
- Describe installation and migration process



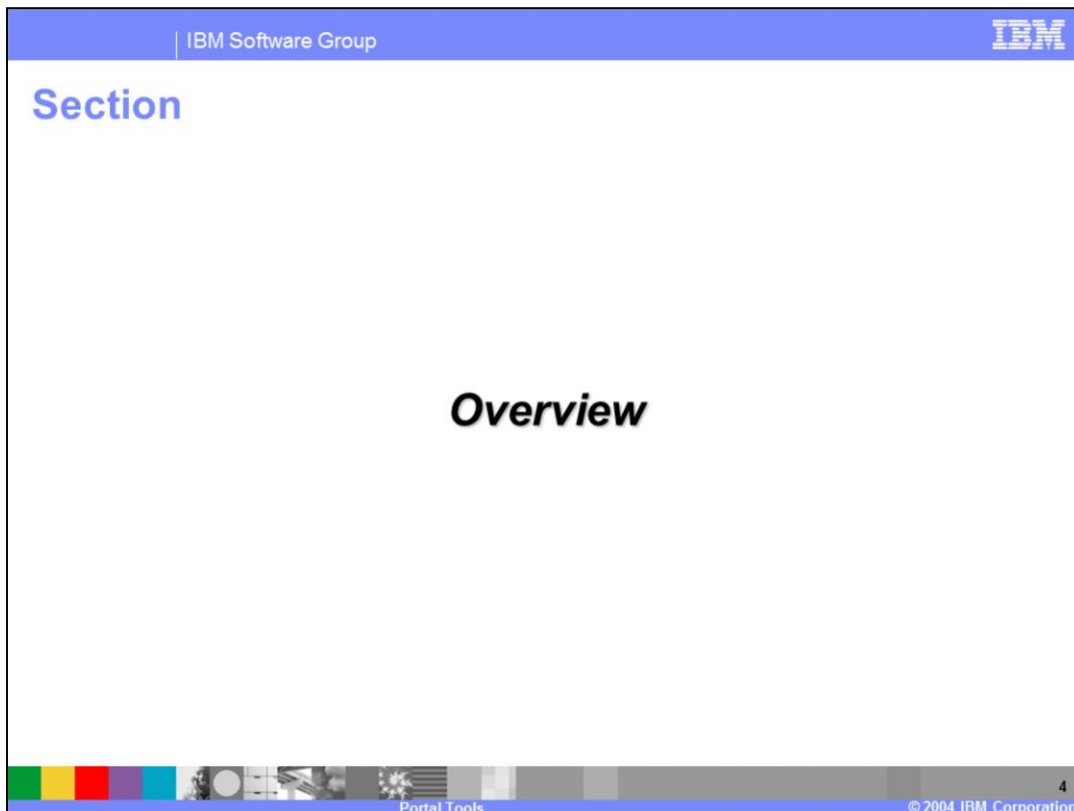
The goals for this presentation are to discuss the installation process for installing the portal tools and test environments for portal development, the migration path for updating to Rational Application Developer v6, and lastly the new features that have been added to the portal tools.

## Agenda

- Overview
- Installation and Migration
- Portal Development Tools
- Advanced Portlet Development
- Testing and Debugging



The agenda for this presentation is to discuss an overview of the portal development tools, installation and migration steps, and then the new features included in the portal tools.



The next section will provide an overview of the portal development tools included with Rational Application Developer.

## Overview

- Portal Tools provide support for portal and portlet development within Rational Application Developer
  - ▶ Portal Toolkit is no longer provided as a Stand-alone install in Application Developer V6
  - ▶ Portal Toolkit now referred to as “Portal Tools”
- Integrated tools provides first class support for portal application development
  - ▶ Development process and tools have been further enhanced over Application Developer V5.1.2
  - ▶ Portlet and portal development support for WebSphere® Portal Server V5.0.2.2
  - ▶ Portlet development support for WebSphere Portal Server V5.1
    - Portal development support coming post-V6.0



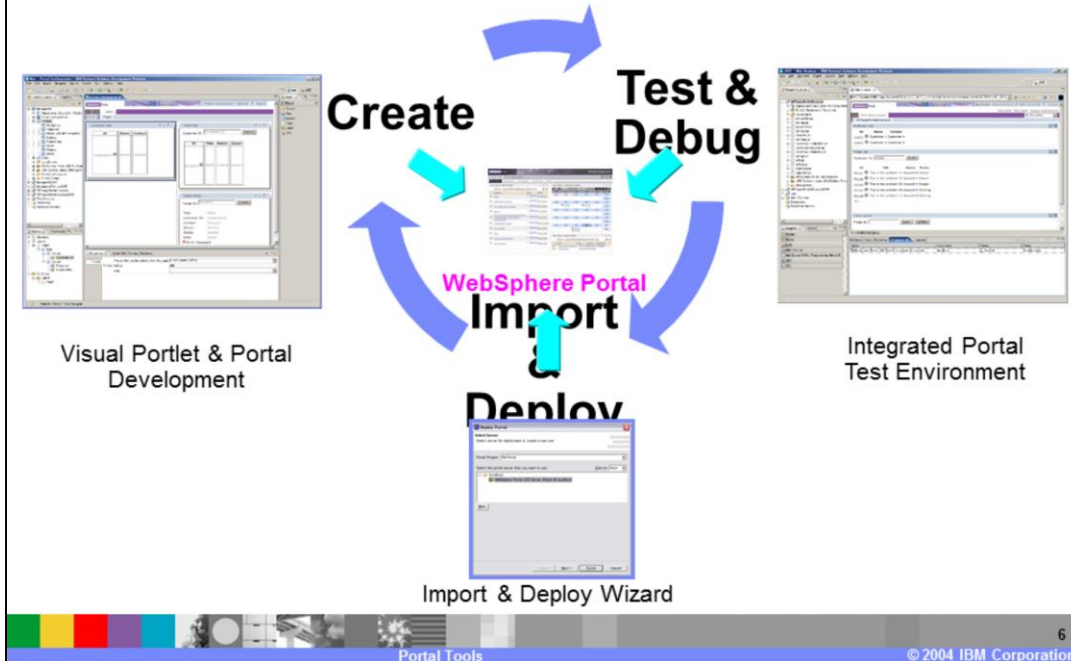
The portal development tools within Application Developer v6 have been expanded to include the ability to develop portals as well portlet applications. The new tools and wizards associated with portal development will be introduced later in the presentation.

Additionally the portal tools are now included with Rational Application Developer as an additional feature that can be installed. This simplifies the install process and makes installing the portal tools more intuitive. No longer do customers have to locate and download the correct version of the Portal Toolkit for a specific version of WebSphere Studio. The inclusion of the portal tools with Application Developer provides a tight integration with the other tools to provide a seamless development environment for portal, portlet, and J2EE development. As a result of the changes described above, the portal toolkit is now referred to as “Portal Tools”.

The portal tools within Application Developer also contain a number enhancements over the previous version of the Portal Toolkit. New development tools have been added that are specific to importing, creating, modifying and deploying a portal from within Application Developer. Also the tools have been expanded to support both the WebSphere Portal Server v5.0.2.2 and v5.1. An important thing to note on this slide is the development types that are supported for the different versions of the WebSphere Portal Server. Application Developer v6.0 supports developing portlet applications for both WebSphere Portal Server

v5.0.2.2 and v5.1. Application Developer v6.0, however, only supports developing portals for WebSphere Portal Server v5.0.2.2. Support for developing portals for WebSphere Portal Server v5.1 will be becoming in a future version of Application Developer.

## 1st Class Support for Portal & Portlet Development



The Portal Development Tools within Rational Application Developer support the complete development lifecycle when developing portal and portlet applications. Customers can use the import or new project wizards to create portals, themes and skins, and portlet applications within Application Developer. Portal tools supports creating portlet applications that use either the IBM or JSR-168 API. Additionally portlet applications can be designed using the basic portlet framework, JSF portlet framework or Struts-portlet framework. After the projects are created within Application Developer, the customer can use the development tools available to design and create their portal and portlet applications. The development tools include a new Portal Designer editor for modifying portals and support for editing themes and skins. During the development process the portal tools are also capable of executing the portal or individual portlet applications against a test environment or a remotely running WebSphere Portal server. After sufficient testing and debugging has been performed against the portal and its portlets, the customer can use the deploy or export wizards to install the portal's configuration, themes and skins, and portlet applications on their portal server.

## Supported Portlet APIs

- IBM API
  - ▶ IBM-defined legacy portlet API
- JSR 168 API
  - ▶ New standard Java™ portlet API
- Separate Portlet Project wizards for each API
- See WebSphere Portal Server V5.0.2.2 and V5.1 support charts for more information



Rational Application Developer supports both the IBM Portlet API and the JSR 168 API. The IBM Portlet API is the original portlet API supported and is specific to IBM WebSphere Portal's implementation. The IBM Portlet API takes advantage of many of the capabilities and services provided by WebSphere Application Server. The other supported API follows the JSR 168 portlet specification. This API has been agreed upon by the industry and defines specific functions and capabilities developers can use when developing portlets in a standard way and specific items which a portal server provider must support. The JSR 168 API is less feature rich as the IBM Portlet API, however it does have the advantages of being based on standards and compatibility with portal servers.



## Supported Frameworks

- Struts portlet
  - ▶ Based on Struts
- Faces portlet
  - ▶ Based on JavaServer Faces (JSF)
- Use Portlet Project wizard and Portlet wizard to create a portlet using one of these frameworks
- See WebSphere Portal Server V5.0.2.2 and V5.1 support charts in Appendix for more information



Rational Application Developer supports developing portlet applications using the basic portlet framework, the Struts portlet framework, and the JSF portlet framework. There are wizards available that help create portlets using each of these frameworks, which you can develop graphically, using the standard Application Developer palettes.

## Advanced Portlet Features

- Cooperative portlets
  - ▶ Broker based data exchange between portlets
  - ▶ Provides declarative programming using taglib and WSDL (Web Services Definition Language)
  - ▶ Portal Tools supports adding Click-to-Action and portlet wiring
- Collaboration portlets
  - ▶ People Awareness (Sametime integration)
  - ▶ Collaborative feature using taglib
  - ▶ Portal Tools supports adding People Awareness feature
- See WebSphere Portal Server V5.0.2.2 and V5.1 support charts in Appendix for more information



Rational Application Developer supports developing portlet applications that use cooperative and collaborative portlet technologies.

The cooperative portlets technology allows portlets on the same page to share and exchange data among themselves. Application Developer supports adding either Click-to-Action or portlet wiring functionality to a portlet. Both of these features are available via the Palette view when working on a JSP file. An example of an application that would use cooperative portlets is an online order tracking application. In this application you may have two portlets, an order portlet for displaying the order details and a customer portlet for displaying customer details related to the order selected in the order portlet. Using cooperative technology the order portlet can be configured to share the order number that has been selected in the order portlet with the customer portlet. This way the customer portlet can be updated with the customer details relating to the order number that has been selected in the order portlet.

The collaborative portlets technology helps people in your organization work together and share information online to achieve their business goals. IBM Lotus software provides many different solutions that help add collaboration based services to your portal. Application Developer supports adding the People Awareness functionality to a portlet. People Awareness can be added to JSP page by dragging and dropping it from the Palette view. The People Awareness features adds the online status indicator to person links, and adds actions to the Person menu.

To see which portlet APIs and frameworks support the cooperative and collaborative features see the appendix at the back of the presentation.

## Section

# *Installation & Migration*



The next section will discuss the installation and migration process for using the Rational Application Developer portal tools.

## Portal Development Tools: Installation

- Portal Development tools now installed as part of Application Developer's installation
  - ▶ Available for selection as an "Additional Feature"
- Portal Toolkit has been eliminated



Portal Tools are now an option under Application Developer's Additional Features section

When installing Rational Application Developer or any other product which includes or is built on Rational Application Developer, the Portal Tools option can now be selected under the Additional Features section as shown in the screen capture in the slide. Selecting this option will place the appropriate plug-ins in the installation location of the product. Incremental installs can also be performed if you do not select the Portal Tools option during Application Developer's initial installation. Including the Portal Tools along with the Rational Application Developer installation, means that customers will no longer have to obtain the Portal Toolkit separately.

It is important to note that the Portal Tools feature only installs the portal development tools that are used to create and modify portals and portlets within Application Developer. If you want to test and debug your portal and portlets within Application Developer, you will need to go through a separate install process for the portal test environment that you require.

## Portal Test Environment: Installation

- Provides a test environment for WebSphere Portal Server within Application Developer
- WebSphere Portal V5.0.2.2
  - ▶ WebSphere Portal V5.0.2.2 and prerequisite WebSphere Application Server images are packaged in Application Developer V6.0 CDs
  - ▶ Installation is accessible via Rational Application Developer V6.0's Launch Pad
- WebSphere Portal V5.1
  - ▶ WebSphere Portal 5.1 CDs are necessary to install Portal Test Environment are bundled with Application Developer V6.0
  - ▶ Installation is accessible through separately bundled WebSphere Portal V5.1 CD
    - Portal installation includes an option to install the server as a test environment
    - See install.html from Application Developer installation for more details



The portal test environment allows developers to test and debug portal and portlet applications within Application Developer. The Rational Application Developer media includes installs for both WebSphere Portal v5.0.2.2 and v5.1.

To install WebSphere Portal v5.0.2.2, an option is available on the Rational Application Developer's launchpad. The install process will install a base version of WebSphere Application Server and WebSphere Portal Server v5.0.2.2 on your system. The binaries for this installation are stored by default in the `\runtimes\portal_v50` directory of Rational Application Developer's install directory.

To install the WebSphere Portal v5.1 test environment, you must launch a separate `install.exe` file available from the WebSphere Portal v5.1 CD's provided with Application Developer. From the separate installer, select the test environment option to install the v5.1 test environment. The install process will install a base version of WebSphere Application Server and WebSphere Portal Server v5.1 on your system. The binaries for the installation will be placed in the directory that you select during the install process. After the v5.1 test environment is installed, additional configuration is required in order for Application Developer to recognize the new test environment. These steps are available within the `install.html` file or Help provided with Application Developer.

## Portal Test Environment: Supported Platforms

Operating Systems	V5.0.2.2	V5.1
Windows™ XP Professional SP 1	Yes	Yes
Windows XP Professional SP 2	Yes	Yes
Windows 2000 Professional SP3	Yes	Yes
Windows 2000 Professional SP4	Yes	Yes
Linux® Intel - Red Hat Enterprise Linux (WS) 3.0	No	Yes
SUSE Linux Enterprise Server (SLES) 9.0	No	Yes
Windows 2000 Advanced Server SP 3	Yes	Yes
Windows 2000 Advanced Server SP 4	Yes	Yes
Windows 2000 Standard Server SP 3	Yes	Yes
Windows 2000 Standard Server SP 4	Yes	Yes
Windows 2003 Standard Server	Yes	Yes
Windows 2003 Enterprise Server	Yes	Yes

The table on this slide shows the different platforms that support the WebSphere Portal v5.0.2.2 and v5.1 test environments. Notice that the test environments are not supported on Linux for v5.0 but are supported on Linux for the v5.1 test environment.

## Coexistence and Migration

- Application Developer V6.0 can coexist with Portal Toolkit 5.0.x on WebSphere Studio 5.x
- Portlet projects created by Portal Toolkit 5.0.2.2 can be migrated into Rational Application Developer V6.0 using one of the following methods
  - ▶ Migrate workspace directly by opening it with V6
  - ▶ Load projects from source code management system
  - ▶ Import projects in ZIP files using Project Interchange feature
- Projects created with a version of the Portal Toolkit prior to V5.0.2.2 need to be migrated through WAR files (using Export and Import WAR file wizards)



Installations of Rational Application Developer v6 with the Portal Tools feature installed and previous versions of WebSphere Studio v5 with the Portal Toolkit installed can coexist on the same system.

Migrating portlet projects created with the Portal Toolkit v5.0.2.2 is very straight forward. There are several different methods that can be used for migration depending on the type of environment you are working in. Your first option is to migrate the workspace by using Application Developer v6 to directly open it. If you choose this method, you should first back up your workspace before the migration process. This way if migration fails or causes an error in your project you can always go back to last known working version. Additionally you can migrate the project using the Project Interchange feature. To perform the migration process all you need to do is import the project within Application Developer v6. Lastly, if you are working in a Source Control Management or SCM environment you can load the projects directly from the SCM system into your workspace.

Migrating portlet projects created with a Portal Toolkit prior to v5.0.2.2 is more involved and requires the manual process that was used migrating projects in early releases of the Portal Toolkit. To migrate these projects, you will need to open up the version of WebSphere Studio and the Portal Toolkit that you used to create the projects and export them into their respective WAR files. After the WAR files have been created, open



Rational Application Developer v6 and import the WAR files into new projects. As the projects are imported, Application Developer will automatically migrate them for you.

## Section

# *Portal Development Tools*



The next section will discuss the features that Portal Development Tools provide for creating and developing portals and portlet applications.

## Portal Development Tools

- Provide productive and flexible tools for building portal sites
  - ▶ Create new portal projects from scratch or by importing existing portal settings and files
  - ▶ Create portal pages and modify layouts of portlets
    - Portal Designer provides portal page customization
    - Page Designer and CSS Designer provide theme and skin editing support
  - ▶ Test and debug portal and portlet projects using Portal Test Environment
  - ▶ Deploy portal settings and files to a WebSphere Portal Server
- Not a replacement for browser based Administration portlets
- WebSphere Portal Server V5.0.2.2 only supported in V6.0
  - Support for WebSphere Portal Server V5.1 coming post-V6.0



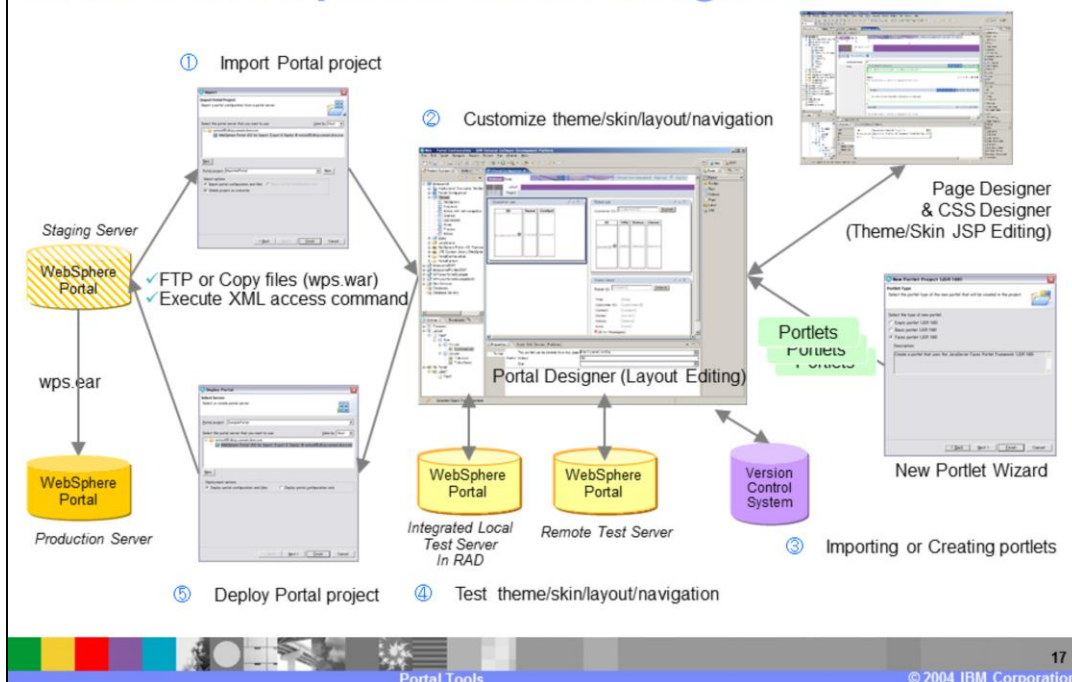
The portal development tools now support the full development life-cycle of creating portals and portlet applications. Through the portal tools, a developer can create a portal project from either scratch or from an existing portal site located on a WebSphere Portal server. After a portal project has been created within Application Developer, the new Portal Designer editor can be used to create or modify the portal. In addition to editing the portlets and portal site navigation structure, Application Developer also allows new themes and skins to be created and existing themes and skins to be edited. When the time comes to test the portal and its associated portlet applications, they can be run on either the portal test environment or on a remote portal server. After the portal has been successfully tested and debugged, Application Developer includes wizards to either automatically or manually deploy a portal, its themes and skins, and its associated portlet applications to a WebSphere Portal server.

Although the portal development tools do provide similar functionality to the Portal server Administration portlets, they are not meant to be a replacement for them. For example, the portal development tools do not provide any functionality for creating user groups or assigning privileges to portlet applications. The Administration portlets should be used instead for this functionality.

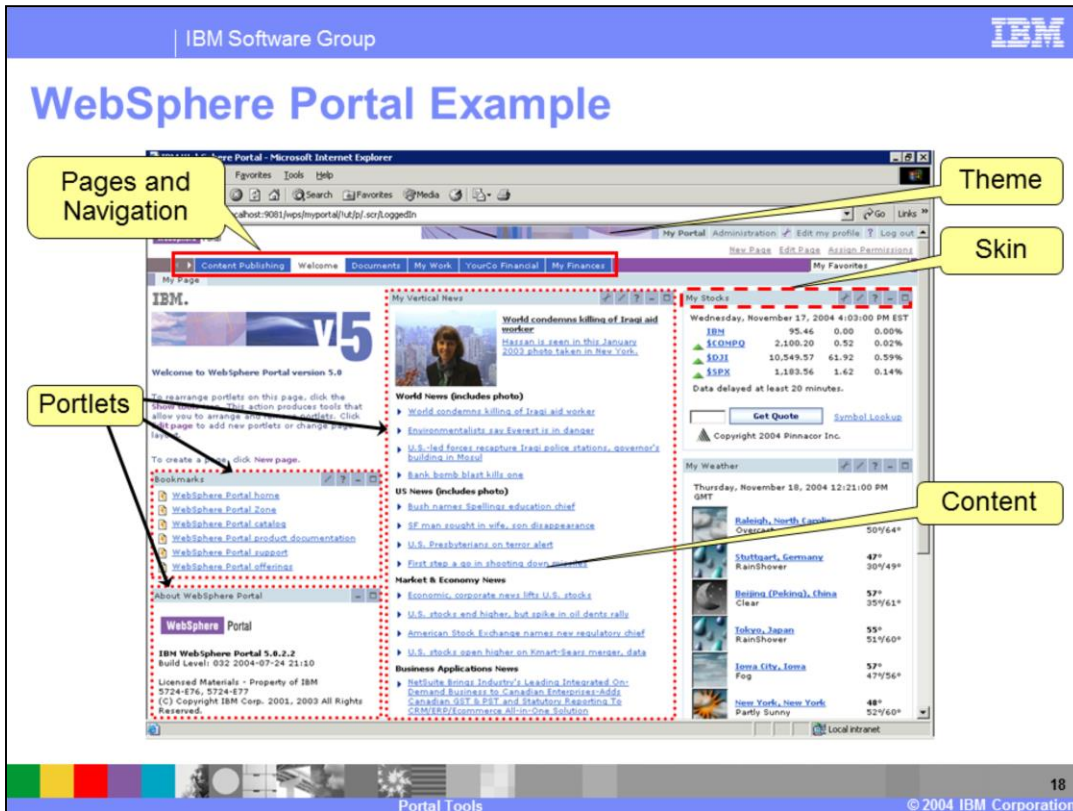
Again it is important to note that the portal development capabilities discussed on this

slide are only supported for WebSphere Portal v5.0.2.2. Support for developing portals for WebSphere Portal v5.1 will be available in a future version of Rational Application Developer.

## Portal Development Tools: Usage Scenario



This diagram summarizes the different development capabilities primarily available for WebSphere Portal v5.0.2.2. Portal applications can be imported into Rational Application Developer where they can then be modified. New portlets can be added to the application and the layout can be established for the portal pages. New portlets can also be created using the various wizards and development tools provided within Application Developer. Portals and portlet applications can be tested and debugged on either a test environment or a remote portal server. To aid in the deployment of a portal to a remote portal server, deploy or export wizards have been provided. The deploy wizard automatically deploys the portal and its associated themes, skins and portlets to the portal server. The export wizard provides a manual way of doing the same thing as the deploy wizard.



This diagram visually describes and shows the different parts that go into constructing a portal. Themes give a portal its overall look and feel. Notice in the slide above it points to the graphics at the top of the page. Along with the graphics, a theme may also determine how the pages and navigation links are displayed within a portal.

Pages and navigation links determine the way in which the portal is organized and navigated. Depending on the selected theme and how the portal designer created the portal, these could appear in many different forms.

Portlets provide the functionality and content to a portal. In the sample above, three of the portlet applications are highlighted. The sample also contains three other portlets which aren't highlighted. These include the My Weather portlet, My Stocks portlet and the Introduction to WebSphere Portal v5 portlet in the upper left hand corner.

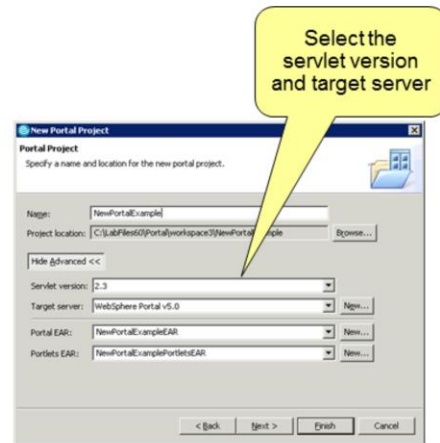
Skins give each individual portlet application its own look and feel on the page. In the sample above, the skin for the My Stocks portlet is highlighted. It is possible to specify a different skin for each portlet application on a portal page.

The Portal Development Tools within Application Developer v6 allows you to modify each

of the different areas defined in the slide above.

## Creating a Portal Project

- Contains resources that effect the structural and stylistic framework of your portal
  - ▶ Themes and skins
  - ▶ Pages and labels
  - ▶ Portal configuration file (ibm-portal-topology.xml)
- File > New > Project > Portal Project



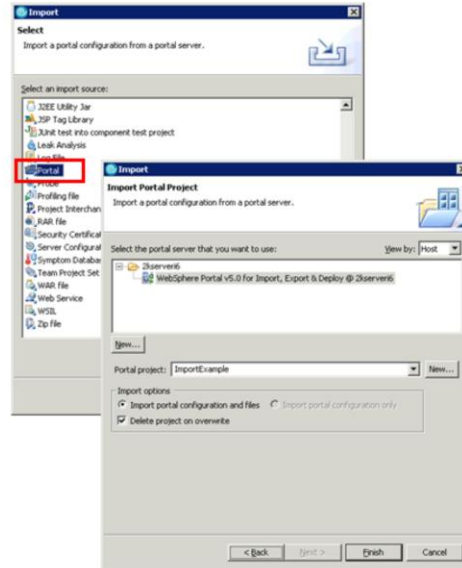
A portal project within Application Developer contains the resources that affect the structural and stylistic framework of a portal. These files include the themes, skins, screen JSPs, and the portal configuration file used to create a portal. The new portal project wizard allows a developer to create a portal project from scratch. The possibility also exists to import an existing portal from a portal server. This will be discussed in an upcoming slide.

The demonstration that is available on this slide takes you through creating a new portal project. To access the demonstration click the Show Me icon.



## Importing Portal Settings and Files

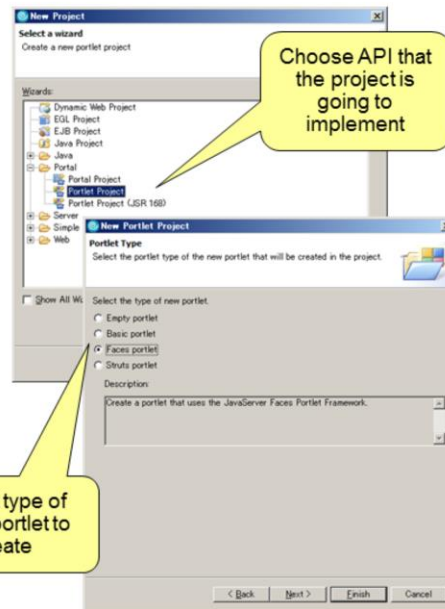
- Import portal settings and files from a remote portal server into a portal project (copy or FTP access required)
  - ▶ File > Import > Portal
- Supports WebSphere Portal Server on Windows and Linux



In addition to creating a portal from scratch, the possibility also exists within Application Developer to import a portal from a portal server. This allows developers to continue developing an already created portal within Application Developer using the new set of Portal Development Tools. In order to perform an import from a server, the developer will need to have either copy or FTP access to the server that they are importing from. Additionally the import functionality is only supported for WebSphere Portal servers running on Linux or Windows.

## Creating a Portlet Project

- Contains resources for portlet applications
  - ▶ IBM Portlet and JSR 168 portlet API supported
- File > New > Project >
  - ▶ Portlet Project (IBM API)
  - ▶ Portlet Project (JSR 168)
- Specify type of portlet to create
  - ▶ Empty
  - ▶ Basic
  - ▶ Faces (JSF)
  - ▶ Struts



A portlet project is created within a workspace to contain the resources used to create one or more portlet applications. The wizards within Application Developer allow a developer to create portlets based on either the IBM API or the JSR-168 API. Wizards are also included for creating individual portlets using the basic portlet framework, Struts portlet framework or JSF portlet framework.

## Portlet Types

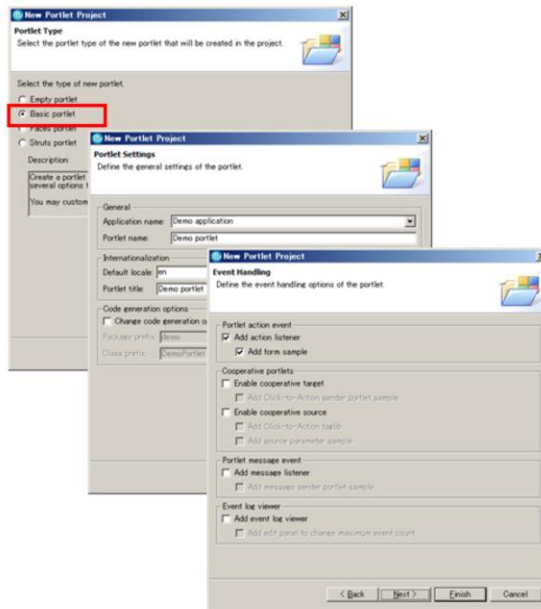
- Empty portlet
  - ▶ A portlet that does not contain any code
- Basic portlet
  - ▶ A portlet that extends the PortletAdapter class
- Struts Portlet (IBM API only)
  - ▶ A portlet that is set up to use the Struts framework
- Faces Portlet
  - ▶ A portlet that is set up to use the JavaServer Faces framework



The portlet creation wizard allows developers to create either an empty portlet, a basic portlet, a Struts portlet or a Faces portlet. The type of portlet that you choose depends on the framework that you are using to implement your application. After you have selected the portlet framework to use and the portlet's configuration parameters within the wizard, Application Developer will handle creating the appropriate sample code and configuration files within your portlet project. For more information on creating a new portlet within your portlet project see Application Developer's help.

## Creating a Basic Portlet

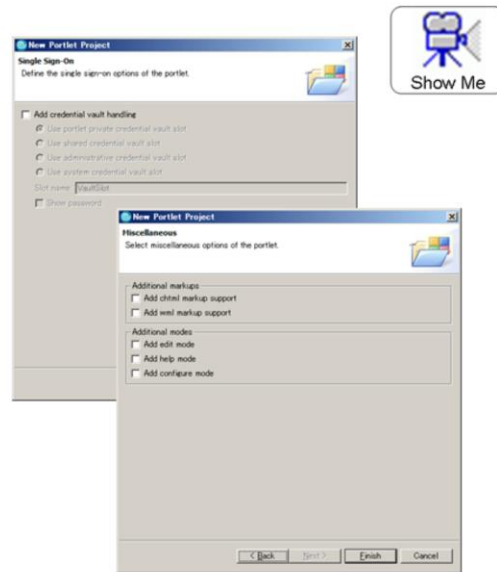
- Basic Portlet wizard
  - ▶ Portlet settings
  - ▶ Portlet application name and title setting
  - ▶ Event handling with sample code
    - Action Event Handling
    - Cooperative portlet
    - Portlet Messaging



The new portlet wizard allows you to configure various aspects of your portlet application. In the slide above, the first few dialogs of the new portlet wizard are shown. The first dialog of the wizard allows you to select the type of portlet you want to create – either an empty portlet, basic portlet or a Struts or JSF portlet. The next panel allows you to modify portlet settings such as the portlet applications name, default locale, portlet title, and the portlet’s package prefix. The third panel allows you generate sample code for portlet action events, cooperative portlets, portlet message events, and event log viewing.

## Creating a Basic Portlet (continued)

- Basic Portlet wizard
  - ▶ Credential vault handling in the Single Sign-On page
  - ▶ Portlet modes
    - Edit
    - Help
    - Configure
  - ▶ Additional markups
    - CHTML
    - WML



The fourth panel allows you to generate sample code to support credential handling for the Single Sign-on page. The last page in the wizard allows you to enable additional support for other markup languages besides HTML and portlet modes. The portlet modes that can be enabled from within the wizard (besides the view mode which is created by default) are the edit, configure and help modes. For more information on the new portlet wizard see Application Developer's help.

The demonstration that is available on this slide takes you through creating a new portlet for a project implementing the IBM API. To access the demonstration click the Show Me icon.

## Portlet Application Limitations

- Use of SDO and Faces Client Components (client-side data caching) in Faces portlet is “prototype purpose only”
- File upload and binary download functions are not supported in Faces portlet
- Creating multiple Struts portlets in a single portlet project is not supported



When creating portlet applications within Application Developer there are a few limitations that a developer should be aware of.

When creating a JSF portlet application, you should be aware that using either the JDBC mediator or client-side data caching technologies within your application should be for prototype purposes only. Applications using either of these features should not be placed within a production environment and will not be officially supported by IBM. When a developer selects to enable either of these features within their project, Application Developer will display a warning to the developer notifying them of the “prototype purposes only” support. Another limitation is that file upload and binary download functions are not support for a Faces portlet application. Lastly, multiple Struts portlet applications can not be contained within the same portlet project. If the need arises to create multiple Struts portlets, you will need to create separate portlet projects to contain each one. Portlet projects can, however, contain multiple Faces portlets or basic portlets along with a single Struts portlet.

## Portal Designer

- Create and modify a portal page's structure, themes and skins
  - ▶ Create portal pages by drag-and-dropping portal objects (labels, pages, URLs, rows, columns and portlets) from Palette view
  - ▶ Change portal page's theme and skins by drag-and-dropping a new theme or skin from Thumbnails view
  - ▶ Change property values for labels, pages and other portal objects through Properties view
- View portal page structure using Outline view
- Launch Page Designer and CSS Designer from Portal Designer for editing portal's themes and skins
- Portal configuration held in `ibm-portal-topology.xml` file in PortalConfiguration directory



The Portal Designer allows a developer to modify either a pre-existing portal imported in from a remote portal server or a portal newly created through the new portal wizard. The Portal Designer allows the developer to modify the portal in a number of different ways. Through the use of the Portal Designer's palette view, portal objects such as portlets, rows, columns, pages, and URLs can be dragged and dropped on a page to quickly and easily modify a portal site. The Portal Designer makes visualizing how each of these different objects are going to look within the portal because it is a What-You-See-Is-What-You-Get or WYSIWYG editor for portals.

The file that Portal Designer is working against and saving changes to is the `ibm-portal-topology.xml` file or the Portal Configuration file and it is stored in the PortalConfiguration directory of a portal project. By default when a portal project is created, the Portal Configuration is automatically opened in the Portal Designer for you. If you need to open it manually it can be opened from either this directory or the by double-clicking on the Portal Configuration entry in the Project Explorer view.

Some of the other views within the Web perspective can also be used during the modification of a portal. The Thumbnails view can be used in conjunction with the Portal Designer to easily change the themes and skins that the portal is using. The Property view allows you to quickly change properties for a selected object in the portal. And the Outline

view provides an alternative to selecting portal objects within the Portal Designer itself.

Another nice feature of the Portal Designer is that it allows you to quickly open Page Designer or CSS Designer with the portal's currently selected theme. To open either of these editors for the theme, right-click within the Portal Designer editor and select Edit Style or Edit Theme.



# Portal Designer

The screenshot displays the IBM Portal Designer interface within the Rational Software Development Platform. The main workspace shows a visual representation of a portal page with several portlets: a 'Customer List' portlet containing a table with columns 'ID', 'Name', and 'Contact'; a 'Ticket List' portlet with columns 'ID', 'Title', 'Status', and 'Owner'; and a 'Ticket Detail' portlet. A 'New Portal Project Wizard' dialog is shown in the background, and an 'Outline' view on the left shows the hierarchical structure of the portal page, including elements like 'Label', 'Page', and 'Portlet'. A 'Properties' view at the bottom shows settings for the selected portlet.

Visualize and customize portal page

New Portal Project Wizard

Apply themes and skins to change appearance

Use Outline view to view portal page's structure

Drag-and-drop a page, label, portlet,...

Portal Designer

Portal Tools

© 2004 IBM Corporation 27

This slide shows that the Portal Designer uses a visual approach to creating a portal. From the slide you can see the number of different views that can be used in the development of a portal.

## Demo: Creating and Testing a Portal

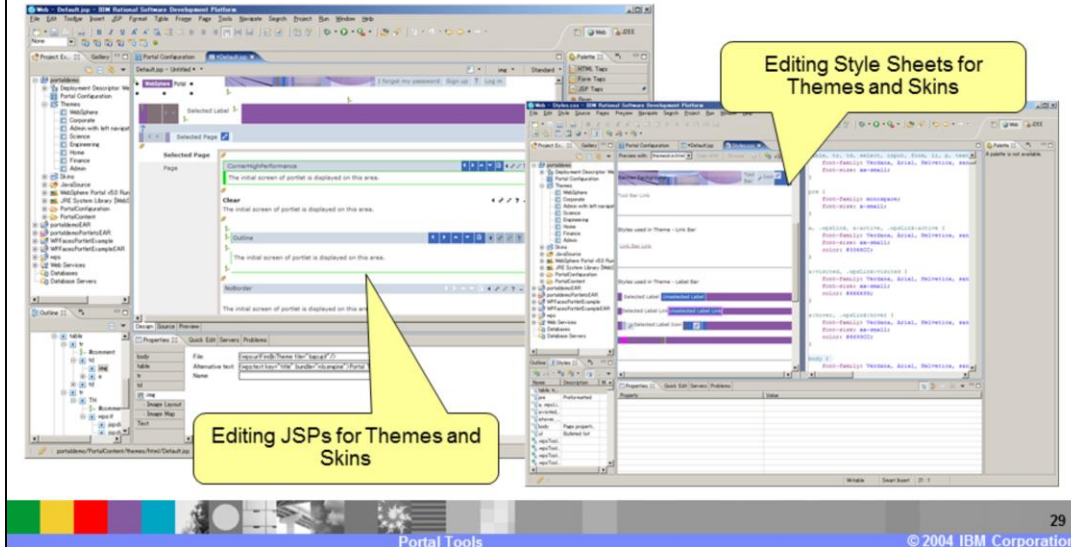
- Create a portal using Portal Designer
  - ▶ Add portlets, labels, pages and columns
  - ▶ Change portal theme
  - ▶ Change portlet skins
- Run and test portal on Portal Test Environment



The demonstration that is available on this slide takes you through using Portal Designer to create a very simple portal. The demonstration shows how the Properties and Thumbnails views can be used to modify different aspects of the portal. After the portal has been created, the demonstration takes you through running the portal on a WebSphere Portal test environment. To access the demonstration click the Show Me icon.

## Page Designer and CSS Designer

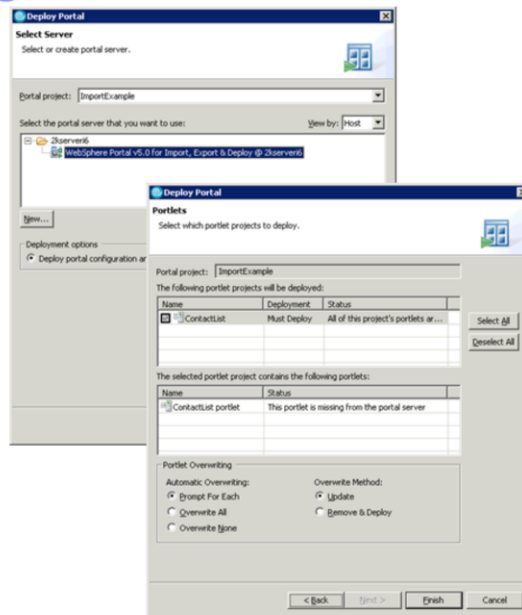
- Page Designer – modify a portal page's theme and skin JSP files
- CSS Designer – modify a portal page's theme and skin style sheet files



As mentioned earlier, Application Developer provides the ability to create new themes and skins as well as edit them. Application Developer provides wizards to create both themes and skins. After a theme has either been created or imported within Application Developer, a developer can use either Page Designer or CSS Designer to modify the theme or skin's JSP and cascading style-sheet files. The slide above shows a theme that has currently been opened in both the Page Designer and CSS Designer.

## Deploying Portal Settings

- Deploy portal project and related portlets to remote portal server (copy or FTP access required)
  - ▶ Right-click on portal project and select Deploy Portal
- Supports WebSphere Portal Server on Windows and Linux
- Access permission must be set by administration portlet after deploy

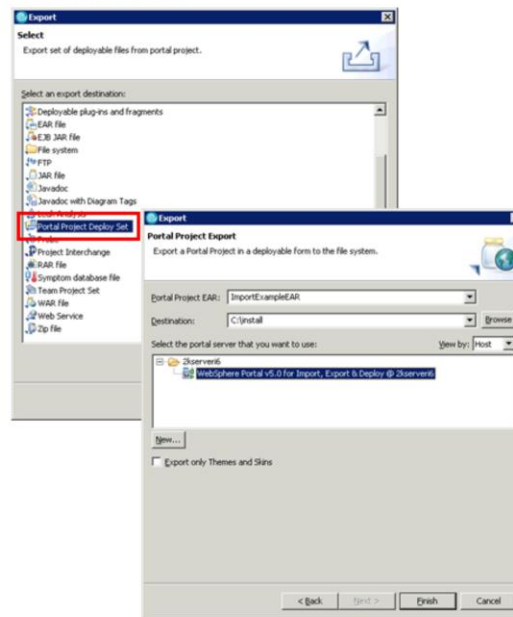


The deploy wizard allows a developer to publish their portal from within Application Developer directly to a WebSphere Portal server. The wizard allows you to deploy only the settings and files that you want to. Through the wizard you have the option of deploying just the portal configuration or the portal configuration and its associated themes and skins. Additionally the wizard allows you to choose which portlet applications that you want to deploy to the portal server. After a portal and its associated portlets have been deployed to a server, access permissions will still need to be set using the administration portlets on the portal server.

In order to perform a deploy to a server, the developer will need to have either copy or FTP access to the server that they are deploying to. The deploy functionality is only supported for WebSphere Portal servers running on Linux or Windows.

## Exporting Portal Settings and Files

- Generates wps.ear, portlet WAR files, XMLAccess scripts for deploying portal configuration and portlets, and deploy instructions file
  - ▶ Manual deployment required
  - ▶ Generated files specific to selected server
- File > Export > Portal
  - ▶ Select a remote portal server to export against
- Access permission must be set by administration portlet after deploy



The export wizard allows a developer to manually deploy a part or all of a portal project to a portal server. Like the deploy wizard, the export wizard allows you to export only the settings and files that you want. Through the wizard you have the option of selecting whether you want to export the portal's configuration, themes, skins and portlet applications. Exporting just the themes and skins, creates a wps.ear file that can be used to register the exported themes and skins with the server. Exporting the portal configuration along with the themes and skins, creates a wps.ear file that is used to register the portal with the portal server. In addition to the wps.ear file, the export wizard also creates WAR files for the installed portlet applications and a XML configuration script that is used to install them. In both situations deploy instructions are generated that contain steps for the deploying the exported files onto a Portal server. After a portal and its associated portlets have been deployed to a server, access permissions will still need to be set using the administration portlets on the portal server. An important thing to note about exported files is that they are server specific. If you want to deploy the set of files on another server, you will need to rerun the export wizard against the new server.

The deploy functionality is only supported for WebSphere Portal servers running on Linux or Windows.

## Section

# *Advanced Portlet Development*



The next section will discuss advanced portlet development features.

## Service Data Object (SDO) Mediators Support

- New SAP and Siebel SDO mediators added to ease Enterprise Application Integration



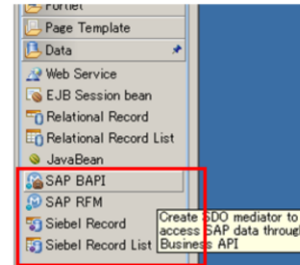
- SDO Mediator support

- ▶ WebSphere Portal Server V5.0.2.2

- JDBC (prototype purposes only)

- ▶ WebSphere Portal Server V5.1

- SAP and Siebel
- JDBC (prototype purposes only)



- SDO Mediators are added to portlets through drag-and-drop from the Palette view and Page Data view



The portal development tools provide access to several different SDO mediators when developing Faces portlet applications. When developing Faces portlet applications for WebSphere Portal server v5.0.2.2, JDBC mediators can be used for prototype purposes only. In Application Developer v6, the ability to use the JDBC mediator within Faces portlet applications has been extended to WebSphere Portal server v5.1. If you want to use the JDBC mediator on WebSphere Portal v5.1, you will first need to download the JDBC SDO mediator from the portal catalog and install it to your Portal test environment installation. The detailed installation instructions for performing the installation will be located on the portal catalog website.

Application Developer v6 has added full support for new SAP and Siebel SDO mediators that can be used to ease enterprise application integration. The new SAP and Siebel mediators are supported only on WebSphere Portal v5.1. The Siebel SDO mediator itself and the associated wizards support the full aspect of Create-Read-Update-Delete (CRUD). SAP systems are method-driven so the operations to perform on them are not your typical CRUD but rather method invocations. All SAP Business API (BAPI) and Remote RFM business functions can be invoked through the mediator. The tooling supports the full aspect of the operations.

As usual the mediators can be added to a portlet through either the Palette or Page Data views.

## Business Process Portlet Development



- APIs available for BPEL (Business Process Execution Language)-based business process portlet development
  - ▶ Process Designers develop business process within WebSphere Studio Application Developer Integration Edition V5.1.1
  - ▶ Portlet Developers
    - Import generated business processes, in the form of a JAR file, within Portlet project
    - Develop and compile task processing portlets using imported business processes
    - Debug portlets using a remote server attach against a WebSphere Portal V5.1 server
- Supports WebSphere Portal Server V5.1 **only**



Task processing portlet

New in Application Developer v6 is the ability to develop business process portlets. Currently there is no tooling support within Application Developer. It is possible, however, to use the APIs to develop your business process portlet. Support for creating business process portlets is only available for WebSphere Portal v5.1.

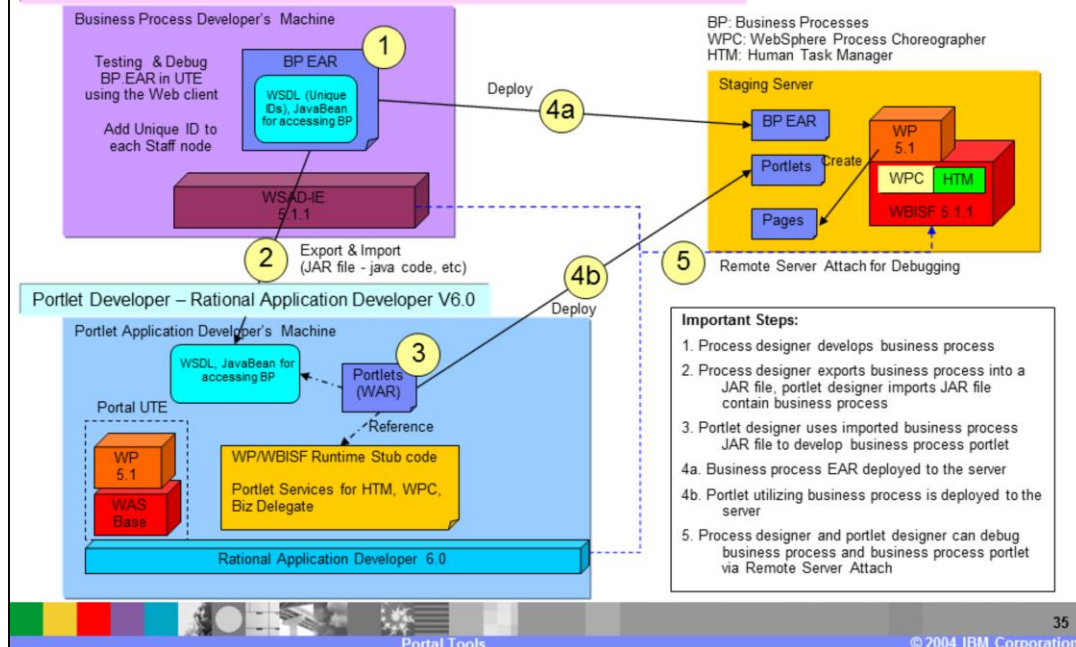
To develop business process portlets within Application Developer, the business process must first be developed within WebSphere Studio Application Developer Integration Edition v5.1.1. After the process designer has created the business process, he or she would then need to export the business process from within WSAD-IE in the form of a JAR file. The portlet developer would then import the exported business process JAR file within Application Developer and use it in the development of the business process portlet. Testing and debugging of the business process portlet would need to occur by using remote server attach to a full installation of a WebSphere Portal v5.1 server. Testing and debugging can not occur on a portal test environment, because the business process portlet requires the WebSphere Process Choreographer. This functionality is only available on full installations of the WebSphere Portal v5.1 server.



# Business Process Portlet Development

New  
V6

Process Designer – WebSphere Studio Application Developer Integration Edition V5.1.1

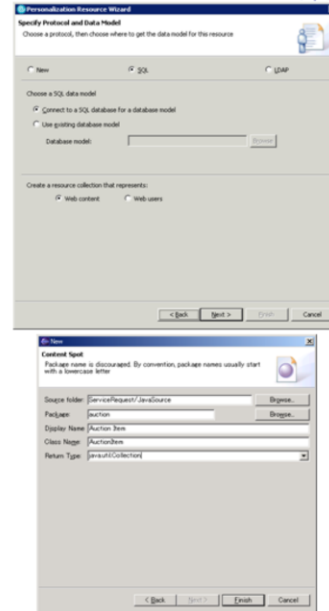


The diagram in this slide further iterates the development process needed for creating a business process portlet. The first step is to create the business process within WebSphere Studio Application Developer Integration Edition. After the business process has been created, the process designer needs to export the business process in the form of a JAR file. The portlet developer will then in turn take the business process JAR file and import it into Application Developer and use to develop a business process portlet. When the business process portlet is ready to be tested and debugged, both the process designer and portlet developer need to deploy their appropriate pieces to the WebSphere Portal v5.1 server. After the parts have been deployed, the business process portlet can be debugged and tested on the server.

## Personalized Portlet Development



- Personalization wizard creates Resource Collections and Application Objects for Personalization Server
  - ▶ Support for database and LDAP as personalized contents resources
- Support for WebSphere Portal Server V5.1 only



The personalization wizard in Application Developer supports creating resource collections and application objects for the personalization server. Resource Collections and Application Objects make up a "Business Vocabulary" to represent the terms and objects upon which Personalization decisions are based. In Application Developer v6, the personalization wizard only supports WebSphere Portal server v5.1.

## Section

# *Testing and Debugging*



The next section will discuss the features within Application Developer that allow you to test and debug portals and portlet applications.

## Testing and Debugging

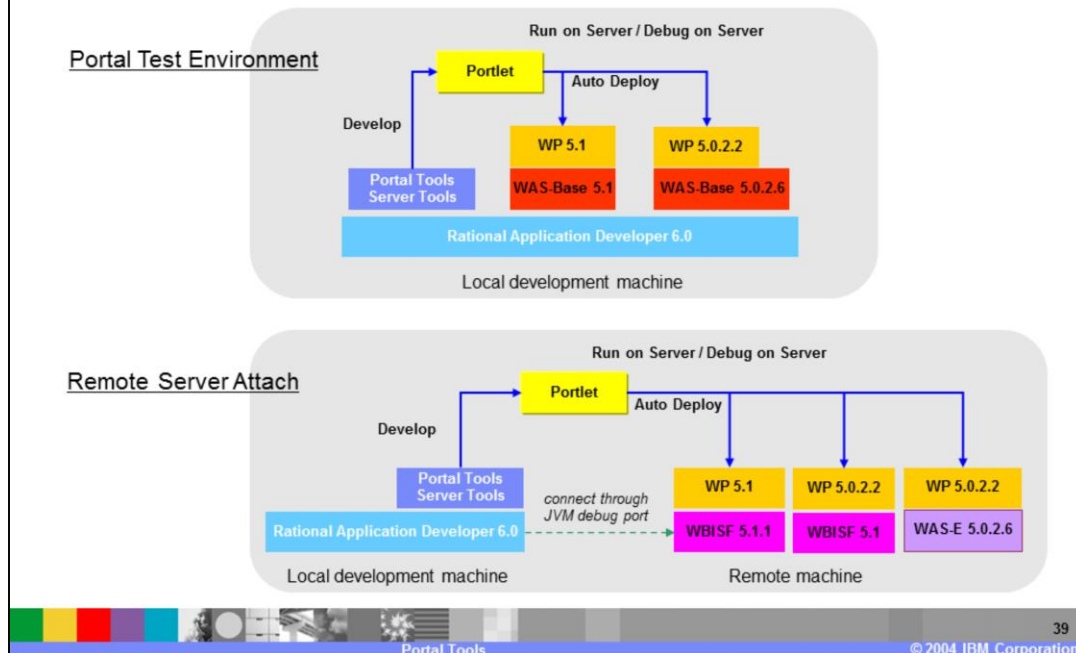
- Testing and debugging supported for portlets or portal pages
  - ▶ Portlet testing and debugging supported on WebSphere Portal Server V5.0.2.2 and V5.1
  - ▶ Portal page (including their associated portlet applications) testing and debugging supported on WebSphere Portal Server V5.0.2.2 **only**
    - WebSphere Portal Server V5.1 support is planned for post-V6.0
- Environments supported
  - ▶ Portal Test Environment
  - ▶ Remote Server Attach
- See WebSphere Portal Test Environment installation table for supported platforms



New in Application Developer is the ability to test both portal and portlet applications. In previous versions of the portal toolkit, developers were only able to test portlet applications. Now with the support for creating and designing portals within Application Developer, you are additionally able to test portals. Portlet application testing and debugging is supported on both WebSphere Portal v5.0.2.2 and v5.1. Support for testing and debugging portals is only supported for WebSphere Portal v5.0.2.2. Support for testing and debugging portals on WebSphere Portal v5.1 will be supported in the next version of Application Developer.

Application Developer provides the ability to test and debug a portal or portlet if it is running in either test environment or on a remote portal server. Past versions of the portal toolkit supported testing and debugging against the test environment only. The ability to test and debug portals and portlets on remote server takes the place of the Portlet Previewer functionality that had been added in the previous iteration of the Portal Toolkit. Portlet Previewer provided developers a way in which they could view their portlet applications on a full installation of a portal server, but did not provide any debugging capabilities.

## Portal Test Environment vs. Remote Server Attach



The diagram in this slide indicates the versions of WebSphere that are installed as part of the different WebSphere Portal installations. Notice that the test environments use WebSphere Base as their base for installation, while the full installs of WebSphere Portal Server (indicated by the Remote Server Attach portion of the diagram) install either WebSphere Application Server Enterprise or WebSphere Business Integration Server Foundation or WBISF as their base. The next slide will discuss situations when you may want to use one environment versus the other.

## Portal Test Environment vs. Remote Server Attach

	Portal Test Environment (Local Debugging)	Remote Server Attach (Remote Debugging)
<b>Portal Location</b>	Local (same machine)	Remote (connected via network, can be same machine)
<b>WebSphere Application Server</b>	Base WebSphere Application Server (no Programming Model Extensions or PME's)	WebSphere Application Server – Enterprise (WAS-E) or WebSphere Business Integration Server Foundation (WBISF)
<b>WebSphere Portal Server</b>	Base functions only	Full portal installed by WebSphere Portal installer
<b>Benefits</b>	<ul style="list-style-type: none"> <li>•Portal server runs against resources in the workspace</li> <li>•Portal will pick up changes in Java and JSPs without publish or server restart (save and run) - Code can be changed while running</li> </ul>	<ul style="list-style-type: none"> <li>•Portlet can be tested on normally installed portal server</li> <li>•You can test portlets using PME dependent functions</li> <li>•You can test your portlet in portal using LDAP</li> </ul>
<b>Restrictions</b>	<ul style="list-style-type: none"> <li>•Functions depend on PME cannot be tested</li> <li>•Portlets using WebSphere Process Choreographer cannot be tested</li> <li>•Portlets are placed on a default portal page</li> </ul>	<ul style="list-style-type: none"> <li>•Every time change is made, export and deploy step must be repeated (Application Developer 6 automates export/deploy, but still user must make explicit action)</li> <li>•Portal + WAS-E/WBISF on debug mode JVM is very slow (both startup and portlet deploy are slow)</li> </ul>

40

Portal Tools

© 2004 IBM Corporation

The table on this slide outlines the key differences between the portal test environment and full installation of WebSphere Portal server. The portal test environment is located on the same machine as Application Developer, whereas, the full portal server could be located on either the same system or across the network. The portal test environment is installed on top of WebSphere Application Server Base, meaning that there is no support for Programming Model Extensions or PMEs. The full portal server is installed on either WebSphere Application Server Enterprise or WBISF. The WebSphere Portal test environment includes support for only base functionality of WebSphere. This means that if the applications you are developing require PME, LDAP, or WebSphere Process Choreographer support you would need to use a full installation of WebSphere Portal. Reasons for using the portal test environment are that changes within the workspace are picked up and applied automatically for you. On a full portal server install, you would need to run the deploy wizard each time you wanted to apply new changes to the remote portal server. Additionally the portal test environment runs faster when running the server is debug mode.

In most cases the test environment is sufficient for testing a debugging your portlet applications. If your application needs functionality provided by a full portal install then it is recommended that you test and debug against this environment.

## Portal Page Test and Debug



- Provides debugging capability to JSP and Java files in a portal project and its associated portlet projects
  - ▶ Portal page's theme, skin and screen JSP files
  - ▶ Portlet JSP and Java files
- Tests portal site's settings for navigation structure, page layout, theme and skins
  - ▶ No debug capability for these settings since they are stored in the portal database
- Avoids unnecessary portal server restart by automatically reloading modified theme and skin JSP files
  - ▶ This function is available only for Portal Test Environment



New in Application Developer v6 is the ability to debug portals. Using Application Developer it is possible to debug the Java and JSP files that make up a portal and its associated portlet projects. The files that you will be able to debug are the portal's theme, skins, and screen JSP files. Additionally you will be able to debug the JSP and Java files that make up a portlet application.

Testing and debugging a portal before deploying it to a production environment also allows the developer to test a portal's navigation structure, page layout and the themes and skins that they have created. These type of visual debugging was only possible in the past, when a portal was deployed to an actual portal server. Now the look and feel of the aforementioned areas can be worked on before the application is actually deployed to a staging or a production portal server.

Remember that if you are testing and debugging on the portal test environment that modified JSPs within your project are automatically reloaded for you. This helps alleviate the number of server restarts that may be needed when making a series of changes to your portal or portlet applications. See Application Developer's help for enabling this same type of behavior on remote portal server.

## Section

# *Troubleshooting*

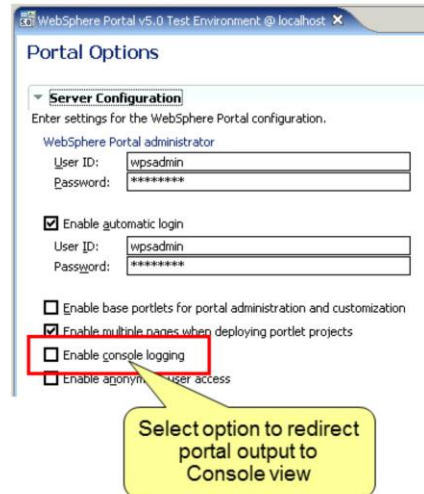


The next section will discuss tips on how to troubleshoot problems encountered when working the portal development tools and portal test environment.



## Viewing Portal Log in Console

- Installation log file of Portal Test Environment
  - ▶ `INSTALL_ROOT\PortalUTE50\log\WpsInst.log`
- Runtime log files of Portal Test Environment
  - ▶ `INSTALL_ROOT\runtimes\portal_v50\log\wps_TIME_STAMP.log`
  - ▶ Option available in Server Configuration editor to redirect the portal server's log to the Console view



The first log to check if you are unable to start a portal test environment, is the test environment's installation log files. Within this log verify that the installation of the portal test environment completed correctly. The install log for the v5.0.2.2 portal test environment is the `WpsInst.log` log file located in the `PortalUTE50\log` directory of Application Developer's install directory.

Another logging option to be aware of is the ability to display the portal server's runtime log in the Console view of Application Developer. When this option is enabled, it appends the portal server's log to the WebSphere Application Server's runtime log that it is running on. This makes it easy for developer to view the runtime messages and exceptions that may be encountered when testing and debugging a portal or portlet. If you choose not to enable this option the runtime logs can still be found in the `<Application Developer install directory>\runtimes\portal_v50\log` directory for the v5.0 test environment and the `<Portal v5.1 install directory>\PortalServer\logs` directory for the v5.1 test environment.

## WebSphere Portal V5.0.2.2 Test Environment

- Possible causes of startup problems
  - ▶ Excessively long filenames are being used
    - Test Environment will not be able to start as a result
    - Check the following names and suitably shorten them if need be
      - Project name, package name, class prefix name, or workspace directory name
  - ▶ Excessively long classpath is being used
  - ▶ Microsoft® Internet Information Server (IIS) or firewall application running on your system
  - ▶ Workspace shared by an administrative user and a domain user



Additional problems that may be encountered when starting up a portal test environment are outlined in this slide. When starting a project on a test environment, you may encounter problems if a project name, package name, class prefix name, workspace directory name or classpath being used is too long. As a result of any of the aforementioned problems, the portal test environment may fail to start. To fix the problem, check WebSphere Application Server's and Portal Server's log files to determine the offending problem. If the problem is a path length or classpath length issue, resolve it by shortening the offending item.

Another source of startup problems is either the Microsoft Internet Information Server or a firewall running on the same system that you are trying to start the portal test environment on. If your system has either of these applications running and you are experiencing startup problems, you may want to disable them one by one to determine which application is causing the problem.

If you are working in an environment where a workspace is being shared by both an administrative user and domain user you may experience startup problems. To alleviate, try creating an unshared workspace and then start the portal test environment to determine if the problem still occurs.

## Section

# *Summary and Reference*



The next section will summarize the points discussed in this lecture.

## Summary

- Portal Development Tools have are now included with Application Developer
- Provides first class portal development solution - allowing for the creation, testing and deployment of portal solutions from within one environment



In summary, this presentation has focused on the portal development tools provided within Application Developer v6. The important points to take away from the lecture is that the portal tools are now provided with Application Developer and are part of its install process. Also the portal development tools have been expanded to support the full lifecycle of developing portal and portlet applications. Through the tools, a developer can now create, modify, test, debug, and deploy a portal and portlets from directly within Application Developer.

## Reference

- WebSphere Portal Documentation Library
  - ▶ <http://www.ibm.com/developerworks/websphere/zones/portal/proddoc.html>
  - ▶ WebSphere Portal Server Information Center
  - ▶ WebSphere Portal Release Notes
- WebSphere Portal for Multiplatforms Support site
  - ▶ <http://www.ibm.com/software/genservers/portal/support/>
- IBM WebSphere Portal V5: A Guide for Portlet Application Development (Redbook):
  - ▶ <http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246076.html?Open>

## Section

# *Appendix*

## WebSphere Portal Server 5.0.2.2

- Support for APIs, framework and advanced features

Portlet API	Framework	SDO for JDBC	Cooperative portlet using wiring	Cooperative portlet using Click-to-Action	Collaborative portlet (People Awareness)
IBM WP API	None	Prototype only	Yes	Yes	Yes
	Struts		Yes	Yes	Yes
	JSF		Yes	Yes	Yes
JSR 168	None		No	No	No
	Struts	No Struts JSR 168 support in WP 5.0.2.2			
	JSF	Prototype only	No	No	No


# WebSphere Portal Server 5.1

- Support for APIs, framework and advanced features

Portlet API	Framework	SDO (*1)	Cooperative portlet using wiring	Cooperative portlet using Click-to-Action	Collaborative portlet (People Awareness)
IBM WPAPI	None	Yes (*2)	Yes	Yes	Yes
	Struts	Yes (*2)	Yes	Yes	Yes
	JSF	Yes	Yes	Yes	Yes
JSR 168	None	Yes (*2)	Yes	No	No
	Struts	Yes (*2)	Yes	No	No
	JSF	Yes	Yes	No	No

\*1) SAP/Siebel only. JBDC is provided for prototype only, not for production use. No IBM service is provided.

\*2) Some coding is required.

 No tools support in 6.0 (support will be provided post-V6.0)  
WebSphere Portal V5.1 supports Struts JSR 168 portlets



## Enabling Transcoding Components

- Transcoding is enabled by uncommenting directives from three files
  - ▶ *INSTALL\_ROOT*/runtimes/portal\_V50/shared/app/config/services/
    - PortletFilterService.properties
    - PortalFilterService.properties
    - services.properties
- All lines are commented out with `#DisableTranscoding`
- Remove comments and restart the WebSphere Portal Server

# Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eLogo/Business	DB2	Series	OS/400	xSeries
AIx	DB2 Universal Database	Lotus	pSeries	zSeries

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 registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.