

Table of contents

About the author	2
Introduction	3
Agenda.....	4
Positioning development tools by type of user.....	5
Rational and WebSphere Studio tools	6
Rational and WebSphere Studio tools	7
IBM WebSphere Development Studio Client overview.....	8
How WebSphere Development Studio Client is valuable	9
WebSphere Development Studio Client packaging	10
Getting started with portlet development	12
IBM Portal Toolkit.....	13
IBM WebFacing Tool	14
IBM WebSphere Host Access Transformation Services V6.0	15
IBM WebFacing Deployment Tool with HATS Technology.....	17
iSeries Access for Web V5R4.....	18
WebSphere Portlet Factory.....	19
Rapid portlet development for WebSphere Portal	20
Increase ROI and overcome development challenges	22
Key features of WebSphere Portlet Factory	23
Seamless integration into the portal infrastructure.....	24
WebSphere Portlet Factory key concepts.....	25
Multipage, complex portlets without coding	27
Robust-integration capabilities deliver SOA.....	28
Service-oriented architecture	29
Unlimited variations without maintenance overhead.....	30
WebSphere Dashboard Framework	31
What is a dashboard?	32
WebSphere Dashboard Framework features	33
WebSphere dashboards	34
Lotus Component Designer	36
Lotus Component Designer	37
Rapid application-development framework.....	38
Early-adopter feedback.....	39
Lotus Component Designer road map	40
Adding support for Lotus Notes and Lotus Expeditor	41
Lotus Expeditor	42
Lotus Expeditor	43
Lotus Expeditor 6.1	44
Lotus Expeditor toolkit.....	45
Summary.....	46
Links.....	47
Trademarks and special notices	48

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 IBM Lotus technologies, such as, Lotus Notes and Domino, Sametime, QuickPlace and 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.



Understanding portal tools

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



© 2007 IBM Corporation

Introduction

Welcome to this online course, entitled “Understanding IBM portal tools.”

Portal technology is one of the fastest growing areas in the IT industry today and IBM® is a leader in providing a broad range of robust tools for developing portlets.

This online course provides portal architects and application developers with basic knowledge of these IBM tools for designing, developing and deploying portals and portlets. You will come to understand the positioning for each of these tools within the greater scope of building a set of comprehensive portal solutions that encompass the IT assets that still reside as a part of the enterprise’s mission-critical applications.

Audience: This course is intended for programmers and nonprogrammers who need to build portlets displaying application-based data in IBM 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, as well as a fundamental knowledge of portals and portlet technology, including the Java™ (JSR 168) and IBM Portlet APIs. Another course, entitled “Developing a portal interface,” (and found at the same Web site as this course) is an ideal starting point for acquiring this basic understanding.

Agenda

✓ Position development tooling

- IBM Rational and WebSphere Studio tools
- WebSphere Portlet Factory and WebSphere Dashboard Framework
- Lotus Component Designer
- Lotus Expeditor

Agenda

This course covers the following subject matters:

1. You will learn about the IBM Rational® and IBM WebSphere Studio set of tools for building portlets that work in conjunction with the IBM WebSphere Portal Server.
2. There is an explanation of the functions that are provided with, and the interactions between, IBM WebSphere Portlet Factory and IBM WebSphere Dashboard Framework.
3. There is a discussion of the functions available with IBM Lotus® Component Designer.
4. The robust capability of IBM Lotus Expeditor is reviewed as a final part of this informative course.

Positioning development tools by type of user

	IBM Rational and IBM WebSphere Studio	IBM WebSphere Portlet Factory and IBM Workplace Dashboard	IBM Lotus Component Designer	Templates and composition tools
Business users			Build simple components	Build applications from templates
Power users		Build simple components		Build templates
Application developers	Special needs	Build complex components		Compose applications
Business Partners, Consultants, alpha geeks	Special needs or performance tuned components	Build complex components connecting to complex systems		Deliver components as part of template

Positioning development tools by type of user

Various development options are available to build portals and their associated portlets. A wide variety of tools can accommodate anyone in your organization, including your business and power users, as well as those who are fluent with portal and portlet development. The factors that you should consider when choosing a development option are as follows:

- Who is building or developing the portlet?
- What is their current skill set?
- How complex of an interface is needed?

Agenda

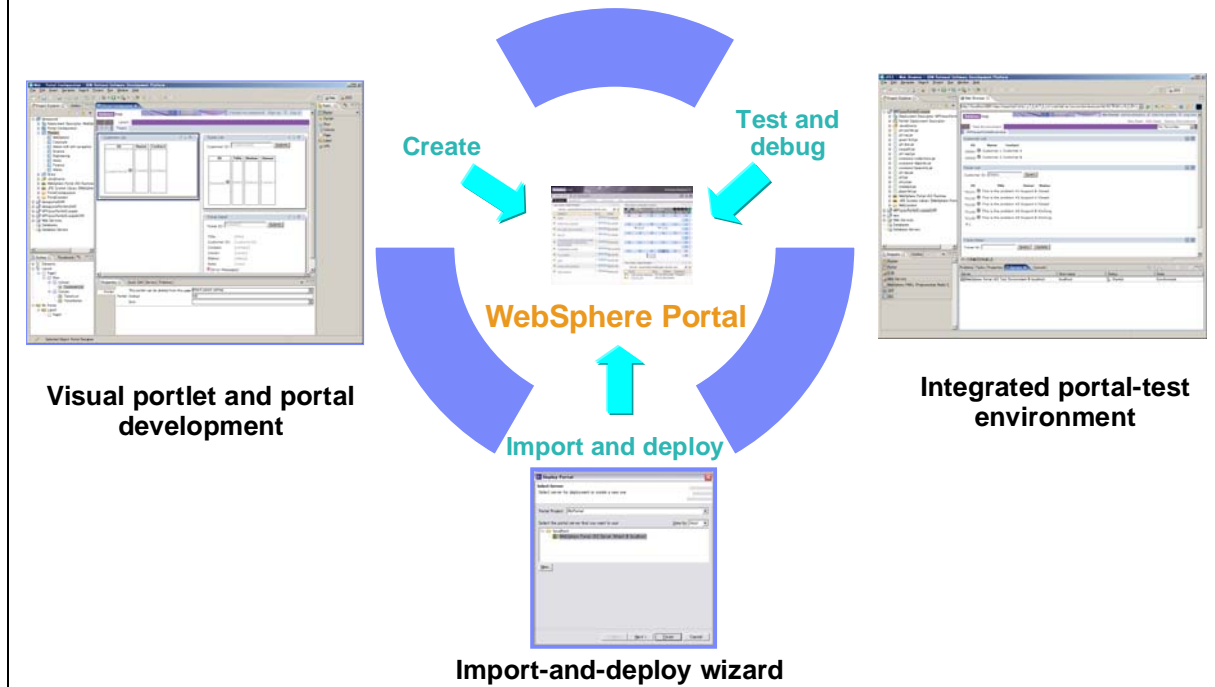
- **Position development tooling**
 - ✓ **IBM Rational and WebSphere Studio tools**
 - WebSphere Portlet Factory and WebSphere Dashboard Framework
 - Lotus Component Designer
 - Lotus Expeditor

Rational and WebSphere Studio tools

The first section of this course provides an explanation of the IBM Rational and IBM WebSphere Studio tools that you can use to develop portlets that run within WebSphere Portal Server.

Rational and WebSphere Studio tools

An integrated portal-development environment for the Java professional



Rational and WebSphere Studio tools

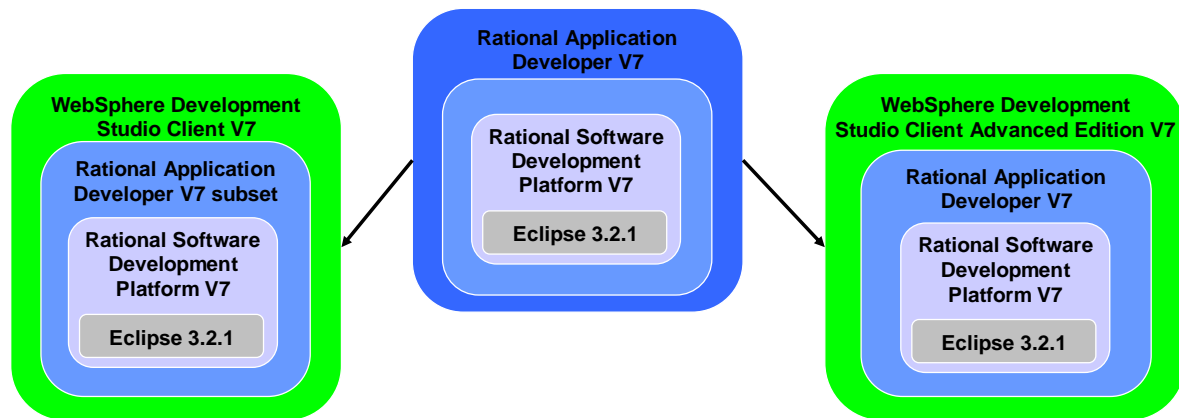
Together, Rational Application Developer, as well as the WebSphere toolset, deliver an integrated portal-development environment for the Java professional to enjoy visually driven portlet and portal development and to, then, deploy the new portlets to users.

These products are fully integrated to support portal and portlet development. No portal toolkit exists on Rational Application Developer V6.0. However, this latest release does have new tools (called *Portal Tools*) that support WebSphere Portal development.

First-class support for portal-application development is also available with enhancements to WebSphere Studio V5.1.2 (and later). For example, WebSphere Portal V5.0.2.2 and WebSphere Portal V5.1 provide a portal-test environment.

As you will see shortly, the WebSphere toolset now provides WebSphere Portal Designer, which is a new tool for customizing a portal page (for example, its layout and navigational links). This is done through the use of visual themes and skins development. There are also import and deployment wizards for building portal pages and the portlets that populate the portal.

IBM WebSphere Development Studio Client overview



IBM WebSphere Development Studio Client overview

IBM WebSphere Development Studio for System i provides a complete toolset for the IBM System i™ application developer:

- The server product includes the RPG, COBOL and C/C++ compilers and the Application Development ToolSet (which includes PDM, SEU, SDA and DFU).
- The workstation tools, called IBM WebSphere Development Studio Client, provide support for building and maintaining native System i applications, as well as for working with Web services and Web and Java applications on the System i platform.

Note: The workstation tools are based on Rational Web Developer, which provides extensive industry-standard support for creating Web applications. WebSphere Development Studio Client Advanced Edition of the product includes all the functions of Rational Application Developer, in addition to advanced System i features, such as support for single sign-on (SSO) and Enterprise Generation Language (EGL).

How WebSphere and Rational work together

WebSphere Development Studio Client for System i is the core WebSphere desktop development tool-set offering. The Advanced Edition includes additional functions to this core product. Similarly, Rational Web Developer is the base development product and Rational Application Developer adds advanced features to it. Therefore, both WebSphere Development Studio Client Advanced Edition and Rational Application Developer are supersets of other products.

WebSphere Development Studio Client for System i is built on Rational Web Developer and adds a set of System i extensions. Similarly, WebSphere Development Studio Client Advanced Edition is built on Rational Application Developer so it contains all the functions of this product plus additional System i extensions. WebSphere Development Studio Client Advanced Edition is a superset of all these products; it contains Rational Application Developer as well as the WebSphere Development Studio Client, which contains Rational Web Developer.

How WebSphere Development Studio Client is valuable

- **Complete tool set helps application developers**
 - Build and maintain System i applications
 - Create Web-enabled front ends to System i business logic
 - Create GUI front ends to System i business logic
- **WebSphere Development Studio Client family built on Rational and Eclipse**
 - * Benefit from Rational investment
 - * Tap into open-source community
 - * Leverage synergy with Business-Partner tools
 - * Enjoy additional skills-based synergy with unified tools

How WebSphere Development Studio Client is valuable

WebSphere Development Studio Client helps you accomplish three primary programming goals from your Microsoft® Windows® workstation:

Goal 1: Develop and maintain System i applications

- Connect to remote systems and perform programming tasks
- Edit, compile, run and debug applications (green-screen, Web-based or combined)
- Work on System i applications when disconnected from the system
- Develop System i business logic in the Java programming language
- Develop in a team-based environment with version control

Goal 2: Create Web-enabled user interfaces to System i business logic

- Develop Java servlets and JavaServer Pages (JSP) files that access System i programs and data, whether your core skills are in Java or the IBM Integrated Language Environment® (ILE) and the original program-model (non-ILE) programming
- Customize the appearance of static HTML pages and JSP files for a Web site
- Test Web applications in a local WebSphere test environment
- Quickly add a Web interface to an existing application and deploy it to a System i application server

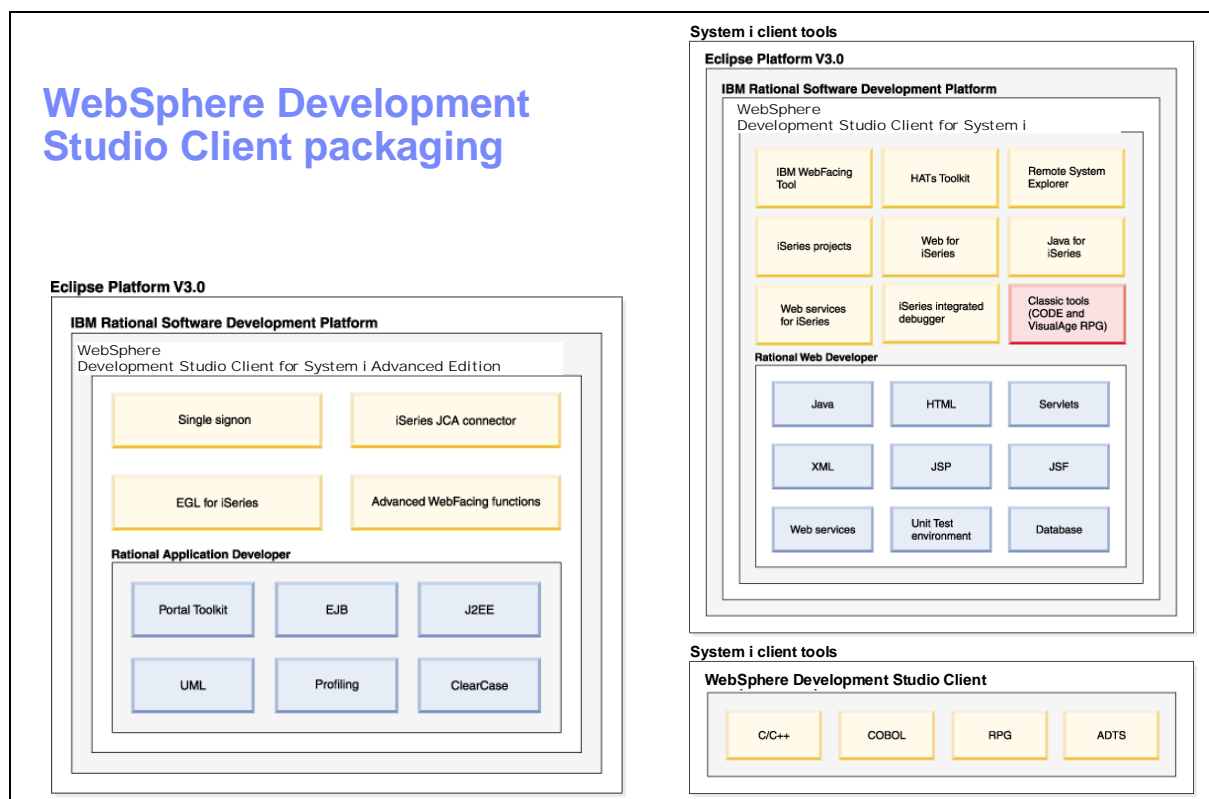
Goal 3: Construct GUIs to System i business logic

- Develop GUIs to RPG programs using visual design tools
- Produce Java applets, Java applications and native Windows applications from the same source

You can accomplish a number of additional development goals using the tools included in both editions of WebSphere Development Studio Client, such as:

- Build XML resources
- Develop SQL statements
- Create Web services
- Generate Web pages from local and remote relational databases

WebSphere Development Studio Client packaging



WebSphere Development Studio Client packaging

When you order WebSphere Development Studio for System i (the server product that runs on the System i platform) you are entitled to receive WebSphere Development Studio Client for System i (the client product). The server and client products include many components, including the following components:

System i server tools

Most System i development environments have a copy of WebSphere Development Studio (5722-WDS) installed on the system. This includes the ILE RPG, COBOL, C, and C++ compilers along with the server-development tools, such as PDM, SEU and SDA.

System i client tools

You can install and run either WebSphere Development Studio Client for System i or the Advanced Edition version on your local workstation. The Eclipse-based integrated development environment (IDE) in both of these products continues to support the importance of developing and maintaining traditional System i applications while also providing integrated tools that enable you to modernize those applications more easily. Although the Advanced Edition offers more capabilities, both share the following foundation and packaging layers:

The **open-source Eclipse 3.0 platform**, with a common public license and royalty-free worldwide distribution, provides a plug-in architecture that enables tight integration with other Eclipse-based offerings from IBM, Business Partners and the open-source community.

IBM Rational Software Development Platform, the core IDE, is the basis of many Rational and WebSphere products, such as Rational Application Developer and both WebSphere

Development Studio Client for System i editions. When you install one or more of these products, Rational Software Development Platform is installed only once; each product integrates seamlessly so all your tools work together in a single IDE. The Rational Software Development Platform provides coordinated installation and services, consistent help navigation, a Welcome perspective and role-based development. This team-unifying IDE has a consistent user interface called the *workbench*, and each product adds its functions to the workbench.

Rational Web Developer, built on Rational Software Development Platform, provides an easy-to-learn IDE that helps you build, test and deploy Web services and Web and Java applications. You can develop drag-and-drop applications with JSFs and Service Data Objects (SDO).

WebSphere Development Studio Client for System i is a tool set for System i developers. It is built on top of Rational Web Developer — when WebSphere Development Studio Client is installed, the Rational product and Rational Software Development Platform are also installed. In addition to the System i tools, you have access to all the Rational Web Developer components through the common IDE.

WebSphere Development Studio Client replaces Application Development ToolSet as the System i developer's default toolset. At its core are Remote System Explorer, IBM WebFacing Tool, and System i Java and Web development tools. Although no longer part of the main installation, the classic System i development tools (CODE and IBM VisualAge® RPG) are included as separately installable CDs.

WebSphere Development Studio Client for System i Advanced Edition offers all the functions contained in the base System i client plus many additional features, such as support for SSO, advanced Webfacing capabilities, EGL for System i, and log and trace analysis. As mentioned, the Advanced Edition is built on Rational Application Developer, so you get sophisticated application-development tools for portal, Java and Java 2 Platform, Enterprise Edition (J2EE) applications.

After you order and install the server and client products, ensure that you follow the recommended workstation-hardware and -software requirements, and that you maintain the latest System i server PTFs and WebSphere Development Studio Client interim fixes.

Getting started with portlet development

- **IBM Portal Toolkit**
 - **For developing portlets**
 - Ships with WebSphere Portal Server Express, WebSphere Development Studio Client V7 Advanced Edition and Rational V7.0 tools
 - **For building portlet applications**
 - A plug-in to WebSphere Studio and Rational tools
 - **For creating, testing, debugging and deploying portlet content**
 - Java Portlet API (JSR 168)-generated portlets
 - IBM Portlet API-generated portlets
 - **For examining sample portlets that demonstrate best programming practices**

Getting started with portlet development

Understanding the details of using the IBM Portlet API and the Java Portlet API (JSR1168) is important, but you do not need to be an expert portlet programmer to do portlet development.

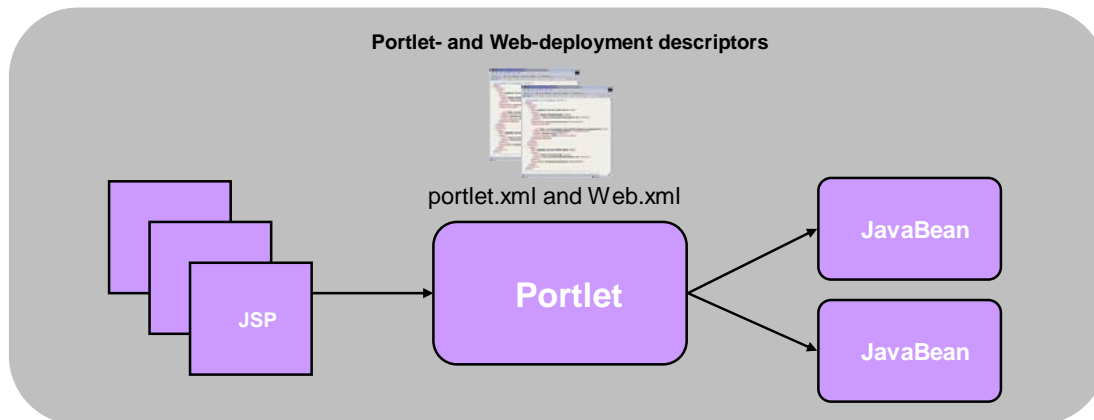
You can develop portlets with IBM Portal Toolkit — a feature available in the Rational and WebSphere Studio developer tools. Portal Toolkit generates portlets and JSPs — the controller and view parts of the application for both the IBM Portlet API and the JSR 168 API. Code generated by the toolkit is a good starting point for customized portlet development.

To learn more about WebSphere Portal Toolkit, visit the following Web site:
ibm.com/software/info1/websphere/index.jsp?tab=products/portaltoolkit

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

IBM Portal Toolkit

IBM Portal Toolkit wizard generates portlet components...



Remaining development:

1. Add business logic to JavaBeans
2. Modify JSP files
3. Make changes to portlet to capture input from JSP files and invoke business logic

IBM Portal Toolkit

IBM Portal Toolkit provides a wizard that generates the basic portlet-application code.

After the wizard has completed, you can modify the generated portlet, JavaBean and JSP files as necessary to call business logic and fine-tune the user interface.

IBM WebFacing Tool

- **IBM WebFacing Tool converts 5250 applications to browser-based applications.**
- **WebSphere Development Studio Client, Advanced Edition V5.1.2 (and later) provides additional capabilities for building portlets from 5250 applications when using IBM WebFacing Tool.**
- **The Webfacing wizard (provided with IBM WebFacing Tool) creates portlets without requiring Java or portlet API programming.**

IBM WebFacing Tool

With IBM WebFacing Tool, you can quickly convert DDS display-file source members so that the user interface for System i programs runs in a browser. IBM WebFacing Tool facilitates the conversion process through user-friendly wizards for selecting the 5250 application's DDS source, converting it and deploying the new Web-based interface to your program as a WebSphere application. JSP and XML files are generated at development time and take the place of your DDS code, thus providing flexibility in customizing the appearance of your new interface prior to run time. Using the style properties provided by IBM WebFacing Tool, you can change Web-page attributes, such as graphics, fonts, colors and layouts. To change a previously converted application, simply reconvert it and select a new style.

WebSphere Development Studio Client Advanced Edition for System i offers the following added capabilities with the IBM WebFacing Tool:

System-screen support: System screens previously had to be Web-enabled using a 5250 data-stream conversion tool and required an interactive workload. System-screen support with IBM WebFacing Tool enables developers to Web-enable entire applications, including UIM-based System i system screens (and other UIM screens), which can then run entirely in batch mode using IBM WebSphere Development Studio Client for OS/400 Standard Edition.

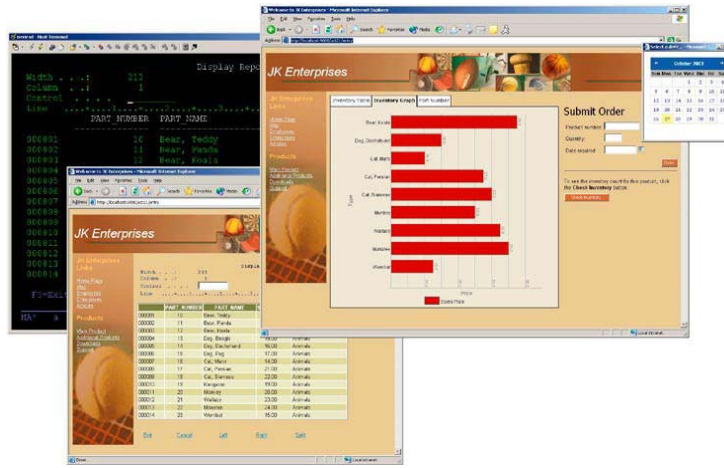
Portal support: IBM WebFacing Tool now allows you to generate Struts-based portlet applications or to add a Webfaced portlet to an existing portlet project that is not based on Struts. You can then deploy these applications to WebSphere Portal Server.

SSO: Enables users to access more than one application and multiple platforms, all by using one user ID and password. For example, you can integrate secure Webfaced and Web-tool-created applications that a user can access without requiring separate IDs and passwords for each.

Cascading style sheets-Positioning (CSS-P): IBM WebFacing Tool supports CSS-P in WebSphere Development Studio Client Advanced Edition for iSeries V6.0.1. This support allows the DDS fields on the Web-enabled interface to be repositioned, unconstrained by the boundaries of the original 5250 area, and offers an almost unlimited ability to customize the Web-enabled interface.

IBM WebSphere Host Access Transformation Services V6.0

- HATS dynamically converts 5250 screens to HTML.
- HATS portlet is created with HATS toolkit – a plug-in to WebSphere or Rational tools.
- HATS toolkit wizard creates a portlet and requires no Java or IBM Portlet API programming.



IBM WebSphere Host Access Transformation Services V6.0

IBM WebSphere Host Access Transformation Services (HATS) Version 6.0 dynamically transforms IBM 3270 and 5250 screens into HTML. You can quickly move applications to the Web without accessing or modifying source code. It provides programmed access to host transactions from standard Web-services interfaces, offers drag-and-drop screen customization, has wizards to customize components and widgets, and supports JSF integration objects for easier Web-page design.

Get on the Web quickly: For new HATS projects, a wizard prompts you for basic information, such as the host address. You then select a project template (one provided by HATS, or your own) or you can work without a template. Next, you can deploy the project to WebSphere Portal or WebSphere Application Server. Finally, you can deliver your host application as HTML to your users' Web browsers. All your screens convert on the fly according to HATS default rules.

Customize at your own pace: Because HATS is rules-based, a set of rules that you create to customize one host screen is easily applied to numerous screens. You can focus on high-traffic and high-value screens; others are converted to GUIs based on the project's default rules. You can use HATS to quickly deploy existing applications to the Web. Later, you can add user-productivity enhancements at your own pace. Because there is no need to customize each screen, HATS continues to work if you make changes to the host application.

Boost productivity, reduce training costs: Training new users on host applications costs time and money. With a little help, host applications that perform mission-critical roles in the organization can continue to be valuable contributors for many years.

Maximize ease of use: HATS can recognize and transform host screens in real time to a Web interface, according to predetermined rules. HATS includes a set of modifiable default rules and offers customizable options (widgets) to transform host-screen elements into HTML-panel components. Widgets include drop-down lists, radio buttons, hot-link lists, button tables and bar graphs. HATS macro support lets you provide programmed navigation through numerous legacy screens. You can combine data from multiple screens and data sources into one HATS screen. HATS macros are easy to generate, allowing you to streamline user interactions with host applications. HATS can also use macros created in IBM WebSphere Host On-Demand.

Improve GUIs: You can add drop-down lists, tables, radio buttons, tabbed folders and other features to host screens to increase user productivity. Users can point and click through the entire host application, including the ability to click PF keys.

Enhance presentations with HTML: You can improve your GUI's appearance by adding HTML components (logos, graphics, pictures and backgrounds). Host applications can have the same look and feel as your Web applications, without sacrificing their value.

Create new applications: You can combine screens and data from multiple host sources with Java technology-based applications to create new WebSphere applications. You can encapsulate transactions with host systems into reusable business objects, such as Web services, JavaBeans or EJBs. You can use HATS and IBM Software Development Platform wizards to create Web pages that call these new business objects; wizards also let you create SQL queries and business objects to implement the queries.

Create Web services: HATS and IBM Software Development Platform simplify the creation of Web-services interfaces that access host applications. Web-services protocols, such as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL), provide reusable support for standardized host access, helping lower the cost of maintaining and deploying connectors to these systems. Web services can integrate host-based business tasks with internal or external applications to augment services-oriented architecture (SOA) solutions.

Integrate robust portal function: HATS can run directly in the WebSphere Portal environment to take advantage of integration with other portlets in the portal. Features, such as click-to-action and cooperative portlets, increase productivity by sharing data between portlets. Credential-vault and Web express-log-on support helps users simplify the process of signing on and providing credentials to multiple applications. HATS works with IBM Tivoli® Access Manager and other vendor products to provide support for Web express log-on.

Optimize security and scalability: HATS offers many security features and leverages the security features in WebSphere Application Server and IBM WebSphere Portal. Secure Sockets Layer (SSL) and secure HTTP (HTTPS) deliver robust security between host applications, the middle-tier server and the user. HATS appears as a typical user to the host application, supporting the use of existing access-security systems, such as IBM Resource Access Control Facility (IBM RACF®). The WebSphere platform supports the workload-management features required for enterprise-class scalability and availability. WebSphere Workload Manager handles load-balancing and failover functions, such as vertical and horizontal cloning. Clustering provides fault tolerance by distributing requests across application servers.

Integrate with the Eclipse development environment: HATS Toolkit is fully integrated in the Eclipse technology-based IBM Software Development Platform, offering an intuitive interface and wizards for customizing screen-transformation rules and responding to changing business needs. Eclipse is an industry-standard, Web-application development environment.

Extensible solution with HATS open architecture: Because HATS is built on industry-leading WebSphere software using the J2EE architecture, HATS has almost unlimited flexibility and extensibility. Using custom Java development methods, you can extend HATS to integrate with other existing systems or J2EE applications. HATS also supports global variables so you can streamline applications and reduce the need for user input. Global variables also provide a means for moving data between applications and databases.

IBM WebFacing Deployment Tool with HATS Technology

- Work in conjunction with the comprehensive tool set in WebSphere Development Studio Client
- Leverage strengths of HATS and IBM WebFacing Tool
- Enjoy ready-to-use integration between Webfaced applications, system screens or non-Webfaced 5250 applications
- Eliminate OLTP charges for HATS applications
- Take advantage of a new rich client

IBM WebFacing Deployment Tool with HATS Technology

IBM WebFacing Deployment Tool with HATS Technology delivers an integrated modernization solution for the System i platform.

Web-enablement all around: IBM WebFacing Tool has been delivering reliable and customizable Web-enabled applications for years. HATS has been providing seamless runtime Web enablement. Now, with the IBM WebFacing Deployment Tool with HATS Technology, IBM offers a single product with the power of both technologies.

Solution for IBM WebFacing Tool users: WebFaced applications can call non-Webfaced applications that are, in turn, dynamically transformed for the Web by HATS technology.

Solution for HATS users: This offering replaces HATS for IBM eServer™ iSeries and HATS for IBM eServer i5. HATS applications created with HATS Toolkit 6.0.4 and deployed to an i5/OS V5R4 system can now connect to the IBM WebFacing Tool server. This eliminates the online transaction processing (OLTP) charge that is, otherwise, incurred on the System i platform. As before with HATS, no source code is required and you can implement programmable screen flows using scripts, variables and macros.

WebSphere Development Studio Client: IBM WebFacing Deployment Tool with HATS Technology works in conjunction with the comprehensive toolset in WebSphere Development Studio Client. Together, they help you create, test, debug, deploy and maintain sophisticated e-business applications with little Java, Web or Web-services programming. The toolset includes the complete HATS Toolkit and IBM WebFacing Tool.

Benefits for the programmer: Previously, Web-enabled applications created with IBM WebFacing Tool in WebSphere Development Studio Client for iSeries were not capable of accessing system screens or non-Webfaced 5250 applications without substantial integration effort. IBM WebFacing Deployment Tool with HATS Technology addresses this interoperability issue. If a Webfaced application encounters system screens or non-Webfaced 5250 applications, HATS dynamic-transformation capabilities are leveraged to deliver a seamless experience in the browser interface. IBM WebFacing Tool V6.0.1 extensions connect to this HATS capability. The system screens or non-Webfaced 5250 applications are transformed based on preset defaults and inherit the same browser style as the Webfaced application.

iSeries Access for Web V5R4

- **Includes several portlets for accessing System i functions and data**
 - 5250 portlet – runs commands and accesses full-screen 5250 character-based applications
 - iFrame portlet – accesses any iSeries Access for Web servlets
 - IFS browsing portlets – browse, view, edit, upload or download files
 - Printers, printer output, and output queues portlets – help manage printer designations, output and output queues
 - Database tables and SQL portlets – view database tables, add and update records and run SQL queries
 - Command portlets – run CL commands

iSeries Access for Web V5R4

iSeries Access for Web offers browser-based access to the System i platform. It enables users to leverage business information, applications and resources across an enterprise by extending the System i resources to the client desktop through a Web browser.

iSeries Access for Web is server-based and is implemented using Java servlet technology. It uses industry-standard protocols (HTTP, HTTPS and HTML) and is lightweight, requiring only a browser on the client. iSeries Access for Web runs batch commands, provides access to the i5/OS database (IBM DB2®), integrated file system (IFS), printers, printer output, messages and a 5250 interface.

A 5250 portlet runs commands and accesses full-screen, 5250-character-based applications. An iSeries Access for Web iFrame portlet accesses any IBM iSeries Access for Web servlets. IFS-browsing portlets browse, view, edit, upload or download files. Printers, printer output and output-queues portlets help manage printer designations, output and output queues. Database tables and SQL portlets view database tables, add and update records, and run SQL queries. Additionally, command portlets run i5/OS command language (CL) commands.

To learn more about iSeries Access for Web, visit the following Web site:

ibm.com/servers/eserver/iseries/access/web

To learn more about iSeries Access for Web portlets, visit the following Web site:

ibm.com/servers/eserver/iseries/access/web/portlets.html

There are links for both of these Web sites in the Links section of this course.

Agenda

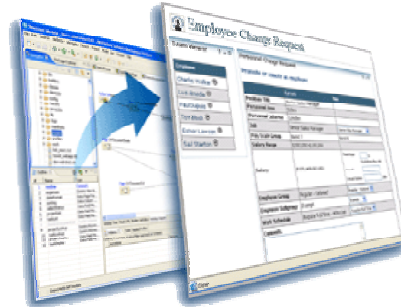
- **Position development tooling**
 - Rational and WebSphere Studio tools
 - ✓ **WebSphere Portlet Factory** and WebSphere Dashboard Framework
 - Lotus Component Designer
 - Lotus Expeditor

WebSphere Portlet Factory

The next section of this course describes the IBM WebSphere Portlet Factory tools.

Rapid portlet development for WebSphere Portal

WebSphere Portlet Factory is a dedicated portlet-creation environment for WebSphere Portal that *simplifies* and *accelerates* the development, deployment, maintenance and reuse of custom SOA-based portlets – including **SAP**, **Domino**, **PeopleSoft**, **Siebel** and **Web-service** portlets.



✓ **Automation** ✓ **Change** ✓ **Reuse**

WebSphere Portlet Factory provides rapid application development and integration to existing applications, data and other IT assets for custom portlet creation - reducing the complexity of J2EE development and speeding IBM WebSphere Portal deployments.

Rapid portlet development for WebSphere Portal

WebSphere Portlet Factory is a tool and technology that enables developers of all skill levels, including IBM Lotus Domino® developers, to quickly and easily create custom portlets for WebSphere Portal and IBM Workplace™. It is an Eclipse plug-in to Rational Application Developer that further enhances WebSphere Portal by providing additional custom-portlet development capabilities. The addition of WebSphere Portlet Factory to the IBM portfolio of portlet-delivery and development capabilities ensures the maximum flexibility, by allowing all portlet developers and contributors within the enterprise to have a role in portal and portlet deployment.

Portlet Factory eliminates the complexity of J2EE development of custom portlets by automating the process through reusable wizard-like components called *builders*. It features very high-level builders, such as those that allow direct SAP and Domino integration, as well as low-level builders that perform even the most rudimentary development tasks. In addition to automating development, builders also ensure consistent and error-free development.

However, the automation provided by Portlet Factory is only one part of its value proposition. The second major benefit is its support for the change and reuse of existing portlets in new contexts — without additional coding — through the use of application *profiles*. Profiles support the dynamic generation of unique portlet instances, which can also be exposed to nontechnical users. In essence, this means the developer can delegate new portlet iteration, creation,

customization and control to business users — again, all without additional coding, thereby eliminating the burden on IT to deliver and manage multiple portal instances.

WebSphere Portlet Factory V6 offers many improvements over the previous release. It generates smaller portlets, while deploying the same functionality. This release also supports the use of application templates, which enables the portlets it produces to work well inside of application templates. There is a new Add Edit Defaults mode that enables administrators to set defaults that users can customize and there is also support for portlet communication, specifically for the use of portlet wiring through a property broker. Additionally, new wizards, tutorials and examples are focused on improved developer proficiency and productivity.

Finally, WebSphere Portlet Factory V6 makes it possible for you to work with more languages, which helps with the demand for national language support (NLS). The languages now supported include: simplified Chinese, traditional Chinese, English, French, German, Italian, Japanese, Korean, Portuguese (Brazilian) and Spanish.

Increase ROI and overcome development challenges

- **Reduce development time = reduced development cost**
- **Increase ROI by realizing new value from existing systems**
- **Expand portlet creation to developers of all skills**

Increase ROI and overcome development challenges

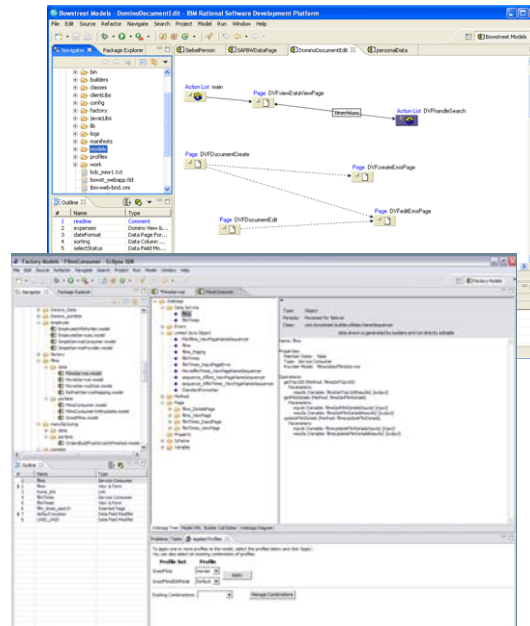
The benefits of using WebSphere Portlet Factory as a portal and portlet development tool are significant. The first key advantage is that development time is much shorter, which not only aids in faster deployment and accelerated business value, but also helps increase developer productivity and reduce portal-development costs.

WebSphere Portlet Factory can help the enterprise increase ROI by leveraging additional value from existing systems. This tool helps you to easily combine data from multiple back-end applications and databases into multipage, role-based, composite-portal applications. You can create, and then reuse, portlet assets from applications that are founded on SAP, Oracle, PeopleSoft, Siebel and Lotus Domino technologies, as well as other enterprise resource planning (ERP) and development platforms.

WebSphere Portlet Factory makes it possible to expand portlet creation to developers of all skill levels. You can develop fully compliant J2EE portlets, regardless of your technical finesse with this complex development environment. You can also use existing IT-development resources more effectively. Finally, WebSphere Portlet Factory provides a path for Lotus Domino, Microsoft Visual Basic, Microsoft .NET, Web developers, as well as other corporate developers to expand their IT assets to J2EE.

Key features of WebSphere Portlet Factory

- Tight integration with WebSphere Portal (automatic-deploy portlets, click-2-action, wiring, people awareness, SSO)
- Plug-in to Eclipse and Rational Application Developer
- Multipage, complex portlets without coding
- SOA delivery and robust integration capabilities with existing applications
- Unlimited customization without the maintenance overhead
- Business-user configuration
- Rapid iteration and change
- Flexible deployment options



Key features of WebSphere Portlet Factory

Numerous features of Portlet Factory ensure the unsurpassed competitive differentiation of WebSphere Portal and contribute to increased portal success and satisfaction for both the enterprise and its customers, employees and partners.

With its tight WebSphere Portal integration and automation through builders, it dramatically simplifies the development of complex portlets — including multipage, transactional composite portlets that integrate with existing applications and systems.

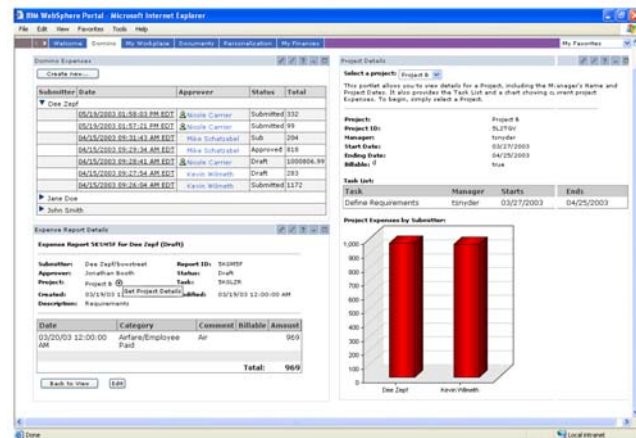
In addition, the patented profiling technology incorporated into WebSphere Portlet Factory generates multiple portlet variations from one code base. This feature enables developers to quickly vary portlets across any number of multiple user dimensions (roles, geography, brands) and greatly simplifies code maintenance by automatically propagating changes. Finally, developers can target multiple computing environments (portal, Web, rich-client, phone, PDA) from a single application, which means you can repurpose applications with minimal development effort.

It is also important to understand that WebSphere Portlet Factory is extensible; it is not limited to the ready-to-go capabilities provided with this tool. For example, J2EE developers can capture an organization's proprietary APIs as custom functions that they can then add to the tool's palette so that other developers can use them. In doing this, they are able to ensure that all developers can access and use the proprietary APIs in the same way, enforcing consistent design patterns and architectures.

Seamless integration into the portal Infrastructure

WebSphere Portal

- Collaboration (people awareness)
- SSO (Credential Vault)
- Portlet-to-portlet communication (click-to-action and portlet wiring)
- WebSphere Portal groups
- WebSphere Portal modes
- Automated deployment



Seamless integration into the portal infrastructure

Some of the points shown on this chart have already been mentioned. However, it is appropriate to be redundant here because the point is that WebSphere Portlet Factory is expressly designed to support seamless integration of all portlets into the portal infrastructure.

Tasks related to collaboration, as well as mission-critical applications, all have the same access point and are tied to an SSO method. It is possible for one portlet to communicate with another portlet through the use of click-to-action and portlet-wiring technologies that are inherent within the WebSphere Portlet Factory tooling.

WebSphere Portal groups and modes are discussed at length shortly.

WebSphere Portlet Factory also supports an automated deployment methodology for delivering new and revised portlets to the portal.

WebSphere Portlet Factory key concepts

One model, driven by multiple profile sets, drives multiple portlets.

Builder

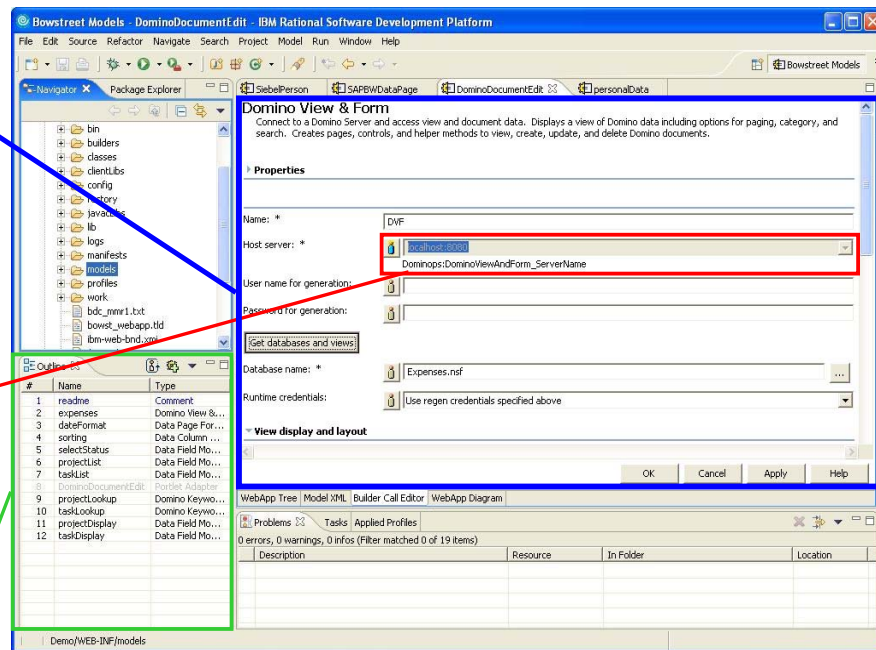
- Wizard-like UI that generates all application code
- Applies developer's modifications immediately
- More than 150 builders are provided with the tool
- Includes the expected portlet function

Profile

- A set of parameters that vary builder inputs to dynamically generate unique portlet versions (personalization)

Model

- Container of builders
- Equivalent to a portlet



WebSphere Portlet Factory key concepts

WebSphere Portlet Factory is such a powerful tool that it is worthy of additional discussion. This course has already referred to the terms *builder*, *model* and *profile*; now you will see how they appear and apply to a developer.

Builder: Builders offer easy-to-use, wizard-like user interfaces that make it both fast and easy to develop portlets. Based on your inputs as a developer, the builder generates all of the application code, including: Java, XML schemas, variables and other components. Builders are also integral throughout the development process, allowing you to go back and change a builder's input values, after which, WebSphere Portlet Factory can immediately update the entire portlet application.

WebSphere Portlet Factory includes more than 150 builders with a wide range of functions. For example, some builders are as simple in purpose as automatically creating the code that places a button on a Web page. An example of a more complex builder is the (Web) service-call builder, which creates the code that calls the specified Web service at run time and stores the output data into a session variable for other parts of the application to access. The service-call builder creates all the necessary variables, XML schemas and other code to call the service and properly map all input and output data into the required SOAP format. Another example is the Domino View and Form portlet builder (shown on this chart). It is a powerful, high-level builder that writes the code to connect directly with the desired Domino data source and to provide a flexible presentation of the data on a Web page.

Model: Builders have to be placed within a portlet container. A *model* is simply a container for an ordered set of builders that, together, automate the creation of a portion of an application.

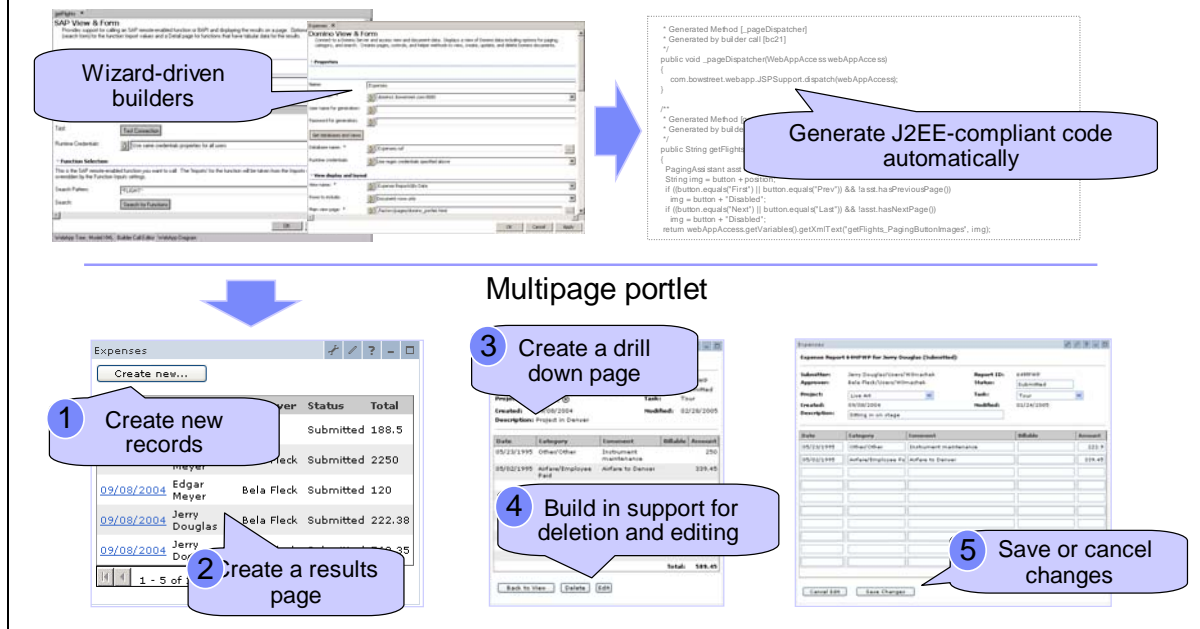
Typically, a model is equivalent to a portlet — the builders that are contained in the model represent the portlet's functions. Models and builders are dynamic. This means that inputs to the builders are easily changed, and a change to the builder input automatically propagates through the entire model. For example, if the database schema changes in this model, the JSP that writes a table in the portlet also changes to reflect the new database schema.

Profile: A profile is a level of abstraction above builders. The input to the builders in a model is captured in profiles. Each profile provides a set of inputs to the builders. When the model regenerates with a specific profile, a new version of the application is dynamically generated. In this way, it becomes easy to generate different behaviors from a single model. For example, an employee-view portlet in a human-resource portal can be designed to show more information to a manager than to an employee; this is accomplished by simply creating a new profile for the employee-view application. Profiles make it possible to reduce the number of portlet-code bases, dramatically improving developer productivity.

Profiles also empower people (who are unfamiliar with the details of your WebSphere Portlet Factory model) to supply input to the model to change the portlet's behavior. For example, you can allow a marketing manager to create customized versions of a portlet that are cobranded and have unique business processes for each new partner. Similarly, you can permit sales representatives to create a new portlet for each new customer that shows only the products that the specific customer is entitled to purchase. Profiles allow you to easily create multiple variations of a portlet from a single code base, dramatically compressing the time and amount of code required to produce variations for different constituencies.

Multipage, complex portlets without coding

* Developers of all skill levels can rapidly build multi-page, complex portlets (search, create, update, and delete) without writing any java code using Builder concept.



Multipage, complex portlets without coding

WebSphere Portlet Factory includes wizard-driven automation components that enable developers of all skill levels to easily build complex, multipage portlets (that provide search, view, create, update and delete capabilities) without writing any code.

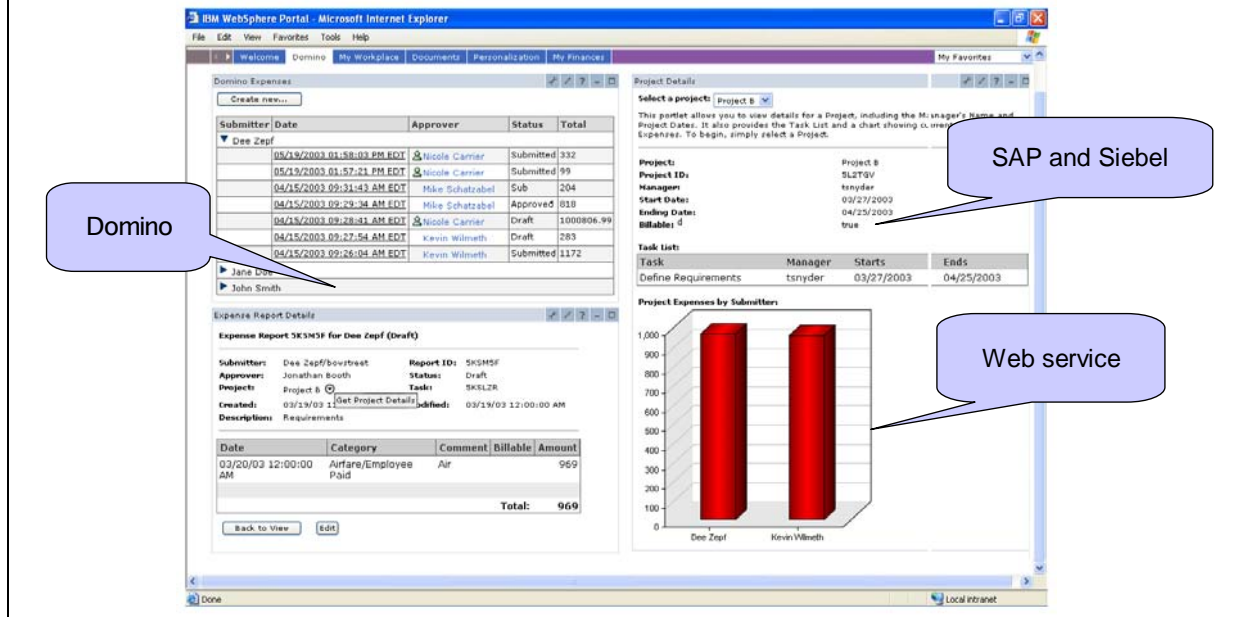
The builders (discussed on the previous page) simplify J2EE coding and application integration complexity through a graphical, wizard-like interface and support the automatic generation of error-free J2EE-compliant code. You can experience more rapid portlet development and testing time frames (as much as 12-fold). And the gains in development productivity do not stop there; as mentioned, WebSphere Portlet Factory can automatically generate new code when inputs change, enabling effortless iteration and routine-maintenance processes.

On the top of this chart, you can see screen captures of two of the more than 150 builders provided with WebSphere Portlet Factory. This example shows how you can use just a few builders (without writing any Java code) to create an entire portlet that does the following actions:

- Reads a Lotus Domino database
- Displays a summary chart
- Enables create, read, update and delete operations

Robust-integration capabilities deliver SOA

Prebuilt extensions or adapters enable enterprises to quickly create composite portlets and applications that leverage data and processes from multiple back-end systems, including: Domino, PeopleSoft, Siebel, SAP R/3, SAP Business Warehouse, Hyperion, databases and Web services



Robust-integration capabilities deliver SOA

As mentioned, WebSphere Portlet Factory includes integration adapters for other applications and IT assets, including the following: Domino, PeopleSoft, Siebel, SAP R/3, SAP Business Warehouse, Hyperion, databases and Web services.

These adapters make it fast and easy to create sophisticated portlets that leverage data from one or more applications and from multiple systems into one portlet application.

Because these adapters automate the process of connecting to the back-end data and building a custom portlet, they eliminate the need to code directly to the vendor's APIs.

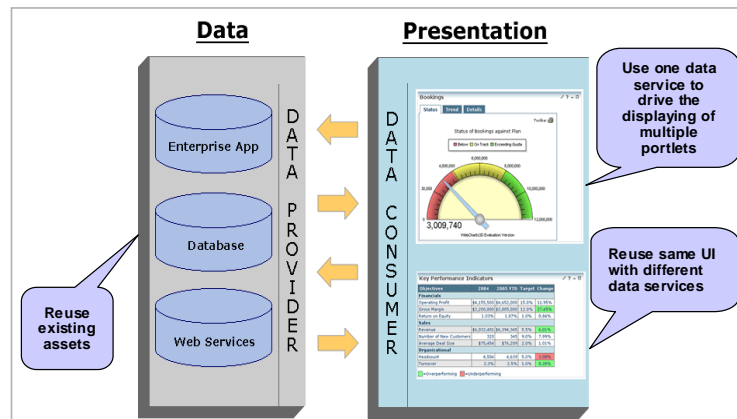
In addition, the WebSphere Portlet Factory data-services layer provides for a true SOA, which allows developers to quickly swap the back-end data source for a portlet without affecting other aspects of the portlet, such as the user interface.

Service-oriented architecture (SOA)

WebSphere Portlet Factory features a **Data-services layer** - providing full support for the service provider and consumer paradigm that is required in an SOA environment. It enables the definition of clean, well-defined service interfaces that separate the user interface from back-end data.

Key benefits

- Enables reuse of existing assets across projects and departments
- Automates back-end connectivity
- Speeds creation of new applications
- Promotes application flexibility



Service-oriented architecture

New data-services builders in WebSphere Portlet Factory V5.12 (and later) provide an entry point to SOA for organizations that do not yet embrace SOA. These new builders also help to further the use of SOA in those organizations that are already on this architectural path. These builders provide a flexible data-services layer that supports your need to create a well-defined interface to separate the user interface from the back-end data. There are a number of valuable benefits to this architecture.

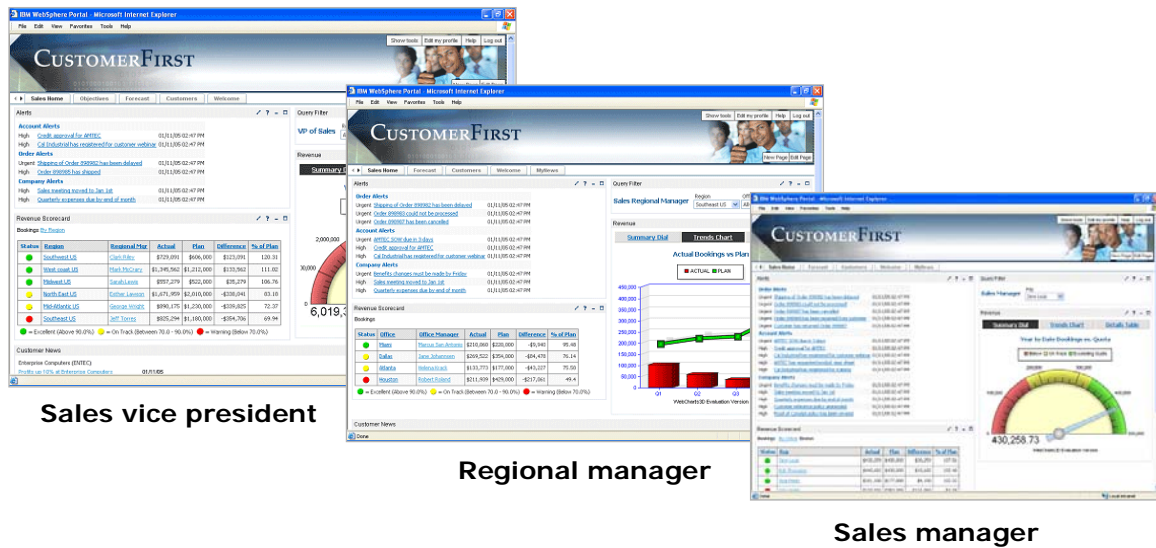
One benefit is that it becomes possible to encapsulate back-end data into a set of precise services that are reusable across enterprise-portal or dashboard projects. The integration builders within WebSphere Portlet Factory can readily consume enterprise applications that are already exposed as services, and can be used to create these services, if they do not already exist.

Another benefit of this architecture is the accelerated creation of new applications and portlets by reusing data services. For example, a data service called *bookings* can drive the presentation of multiple portlets that display bookings data. In the screens shown here, a service is reused to create a portlet with a status gauge of bookings as well as another portlet that shows a table of that same bookings data.

The final benefit of this type of architecture is that it promotes application flexibility. You can reuse the same portlet user interface to build other data services. For example, you can easily swap the bookings data service that drives these portlets, with another data service (such as opportunities). You can see that SOA promotes flexibility by supporting the dynamic swapping of different services, without having to change the user interface.

Unlimited variations without maintenance overhead

Create highly customized portals with the patented automation and profiling technology that is provided in WebSphere Portlet Factory. For example, portals can adapt their structure, presentation, flow, and access rights depending on the role and personal preferences of the user, without any additional coding by reusing existing portlet models.



Unlimited variations without maintenance overhead

After creating a portlet, you can customize it, virtually limitlessly, without additional coding and the maintenance overhead for the numerous portlet-code bases. The dynamic profiling function within WebSphere Portlet Factory enables companies to effortlessly build highly customized portals that contain portlet variations, each tailored to multiple audiences (for example, user roles, partner types, regions, languages and brands). Without profiling, companies must maintain multiple versions of their portlets or write conditional logic to achieve the same effect.

As mentioned, dynamic profiling is another key feature of IBM WebSphere Portlet Factory and WebSphere Dashboard Framework technology. Essentially, profiling solves the maintenance and scalability challenges that companies face when attempting to build customized dashboards that are tailored according to the personal preferences, roles, groups, geography and brand for each user. Using traditional coding methods to write conditional logic or to create a number of slightly different versions is usually cost-prohibitive and, ultimately, a maintenance nightmare.

With profiling, you can very easily build portlets that automatically adapt and tailor themselves according to the preferences and characteristics of the portal user. In the example shown here, a sales dashboard is tailored for three distinct roles. Each time a user logs in, the dashboard automatically adapts to show the pertinent data, processes and features that relate to the user's context and role. Thus, by leveraging profiling, there is no constraint to create a single dashboard that anticipates all users' needs and considers their various technology, role, authority or skills limitations.

Agenda

- **Position development tooling**
 - Rational and WebSphere Studio tools
 - ✓ WebSphere Portlet Factory and **WebSphere Dashboard Framework**
 - Lotus Component Designer
 - Lotus Expeditor

WebSphere Dashboard Framework

This section of the course explains what a dashboard is and describes the functions that are included in the IBM WebSphere Dashboard Framework tooling.

What is a dashboard?

- A user interface that provides a visual representation of **measurements or status**
- An instrumented view of what is occurring **right now**
- A means of providing indicators and alerts
- A conduit for enabling action
- **Not just a scorecard**

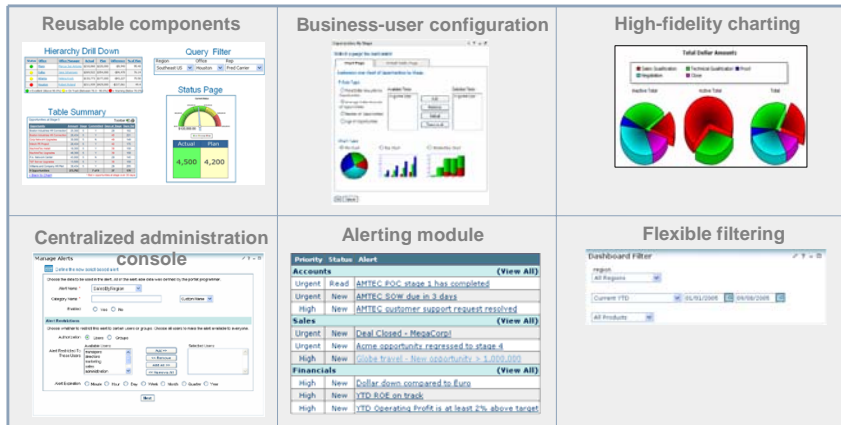


What is a dashboard?

A dashboard refers to an interface that integrates data from a variety of sources and provides a unified display of relevant and in-context information, usually in summary format. Portlets are the vehicle for building the summary information and a portal is the means of displaying a dashboard. A dashboard also provides a means of taking additional actions based on the displayed information. In other words, a dashboard is more interactive than just acting as a means to view a miscellaneous gathering of information.

WebSphere Dashboard Framework features




The Dashboard Framework is a flexible and powerful set of reusable assets, administration tools and dashboard-specific features that speed the creation of custom dashboards.



WebSphere Dashboard Framework features

At a high level, the WebSphere Dashboard Framework includes a set of tools, reusable components and specific features that speed the creation of custom dashboards. For example, WebSphere Dashboard Framework includes a robust alerting engine, sophisticated interactive-charting and -graphing capabilities, filtering functions and other robust features.

WebSphere dashboards

<p>WebSphere Dashboard packages</p> 	<p><u>Available portlet packages</u></p> <ul style="list-style-type: none"> ▪ IBM Workplace Dashboard for Executives ▪ IBM Workplace Dashboard for Sales 	<p>Ideal when enterprises need predesigned dashboard portlets that are capable of being tailored to a specific function and industry</p>
<p>WebSphere Dashboard Framework</p> 	<ul style="list-style-type: none"> ▪ Alerting ▪ Charting ▪ Reusable design components ▪ Business-user configuration ▪ Common administration 	<p>Ideal for enterprises that need a composite framework specifically designed for building dashboards</p>
<p>WebSphere Portlet Factory tooling</p> 	<ul style="list-style-type: none"> ▪ Rapid application development ▪ SOA ▪ Integration extensions ▪ WebSphere Portal adapter ▪ Profiling engine 	<p>Ideal for enterprises that need to build portal-based composite applications and want maximum flexibility</p>

WebSphere dashboards

There are three layers to the WebSphere Dashboard Framework product suite. These layers appeal to different types of enterprises, based upon their profile.

For example, if your company is interested in a predesigned and ready-to-go dashboard solution that is tailored to a specific function and industry, then you might be an ideal candidate for one of the IBM dashboard portlet packages:

- IBM Workplace Dashboard for Executives
- IBM Workplace Dashboard for Sales

Each of these dashboards is built on top of the WebSphere Dashboard Framework and the WebSphere Portlet Factory tooling. As a result, the dashboards take advantage of the features in these underlying technologies, such as native-portal and multisource-data integration.

For companies that need to build, rather than buy, a dashboard, WebSphere Dashboard Framework is the solution. This composite framework sits on top of WebSphere Portlet Factory and is designed for building tailored dashboards. WebSphere Dashboard Framework includes a set of powerful features that are integral to dashboards, such as alerting, charting and filtering.

If your company wants a general-purpose tool that speeds the development of portal-based composite applications, consider WebSphere Portlet Factory. As explained earlier, this tooling streamlines the creation, customization and deployment of industry-standard portlets and allows the rapid assembly of custom portal solutions, such as partner and supplier extranets or employee- and manager-focused self-service applications.

For more detailed information on the WebSphere Dashboards, review the IBM presentation that is available at the following Web site:

ibm.com/be/pdf/en/events/symposium/portal_v6_and_dashboard.pdf

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

Agenda

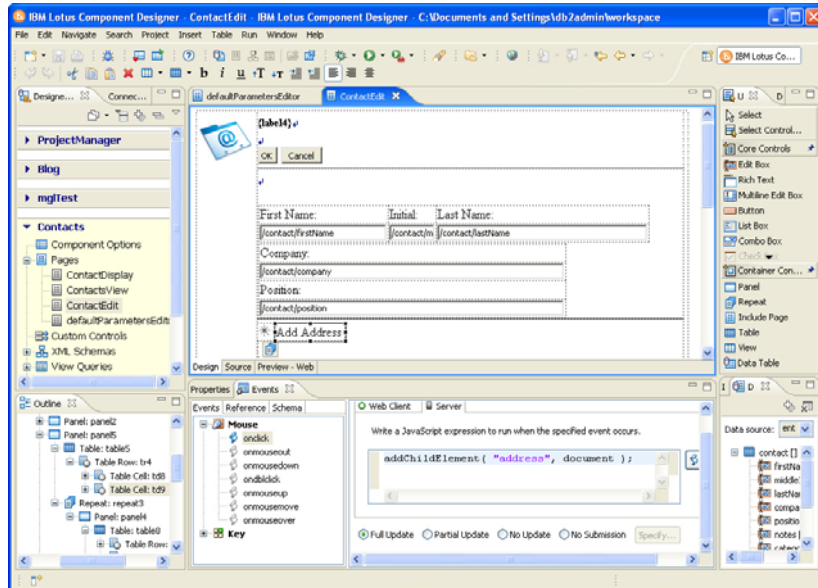
- **Position development tooling**
 - Rational and WebSphere Studio tools
 - WebSphere Portlet Factory and WebSphere Dashboard Framework
 - ✓ **Lotus Component Designer**
 - Lotus Expeditor

Lotus Component Designer

This section of the course itemizes the capabilities of IBM Lotus Component Designer and also discusses its continued evolution as IBM continues to focus on this powerful and popular collaborative platform.

Lotus Component Designer

- **Standards-based (Eclipse 3.2) development tool (formerly Workplace Designer) targets small and medium businesses**
 - Script, Visual Basic and Lotus Domino application designers, as well as others
- **Quickly develop robust components and portlets for WebSphere Portal V6 without writing code**
- **Powerful features for secure, flexible document-based applications (expense reports, document approvals and change requests) or collaboration tools (blogs or discussion forums)**
 - Quickly and efficiently integrate new technology and support for Web 2.0 concepts (AJAX support)
 - Use tutorials and sample applications to get started



Lotus Component Designer

IBM Lotus Component Designer is an innovative, easy-to-use standards-based development tool that script developers, Visual Basic developers, Lotus Domino application designers and others can use to easily create components for IBM WebSphere Portal without writing code. For instance, it is possible to quickly build applications, such as expense reports, project management, document approvals or change requests. It is equally easy to rapidly create customized collaboration tools such as blogs and discussion boards.

Lotus Component Designer is built on the Eclipse 3.2 platform, which makes it is even easier to produce applications, integrate new technology and support Web 2.0 concepts more quickly and effectively. You can more easily bring together people, applications and business processes to support specific job functions through composite-application development. You can quickly create situational applications that serve short-term business needs or extend existing applications with collaborative components.

To find more information about Lotus Component Designer, visit the following Web site:
ibm.com/lotus/componentdesigner

To learn about developer resources for Lotus Component Designer, visit the following Web site:
ibm.com/developerworks/workplace/products/designer

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

Rapid application-development framework

Lotus Component Designer

- **Pages, views, documents and JavaScript**
 - Flexible XML-based document model
- **Separation of user interface and data**
 - Define data models using XML-schema definition (XSD) schemas
 - Define pages visually
 - XML-scripting pages (XSP) markup is an extension to JSF files
- **Parameterization versus coding**
 - Set of high-level components
 - Simple actions for event handling
- **Easy deployment to target platforms**
 - Very fast deployment: Allows incremental development
 - Automatic database setup
 - Manual deployment when necessary

Rapid application-development framework

Lotus Component Designer provides a rapid application-development framework for creating pages, views, documents and the JavaScript that you can use to drive functions within these collaborative components. There is also support for an XML-based document model.

Like other well-designed application-development frameworks, Lotus Component Designer supports the separation of the user interface and the data. You can define data models by using XML-schema definition (XSD) schemas. And Lotus Component Designer is a visual tool, which means you can define pages visually. Internally, this toolset uses XML-scripting page (XSP) markup to build the pages (XSP is an extension to JSF files.)

Lotus Component Designer uses parameters as much as possible (in lieu of writing code). Parameterization allows the tool to generate a set of high-level components that invoke simple actions for event-handling purposes.

Lotus Component Designer supports the easy and fast deployment of collaborative applications to targeted platforms. It also honors the very real need (in most enterprise environments) to allow incremental development efforts. There is the ability to automatically set up a database. Additionally, you can manually deploy the application when necessary.

Perhaps the most important message to deliver in regard to Lotus Component Designer is that there is no need to know the intricacies of the server and J2EE. This rapid application-development framework takes care of that complexity for you.

Early-adopter feedback

Matt Smith – E2 Consulting Group

"Lotus Component Designer provides E2 with the ability to use one tool, built on Eclipse, throughout the full project life-cycle."

"Lotus Component Designer is intuitive enough for business consultants to use; and in the hands of a developer, powerful enough to deliver deep application functionality to the portal."

Alex EL HOMSI – Trilog Group

"Developing project management components that deliver project-centric teamspace functionality using the new composite-application model offered in WebSphere Portal V6."

"'Lego'-like, visual-development model speeds up portlet development by at least 10 times, compared to traditional J2EE development."

"The development model and constructs are very familiar to our Domino developers."

Early-adopter feedback

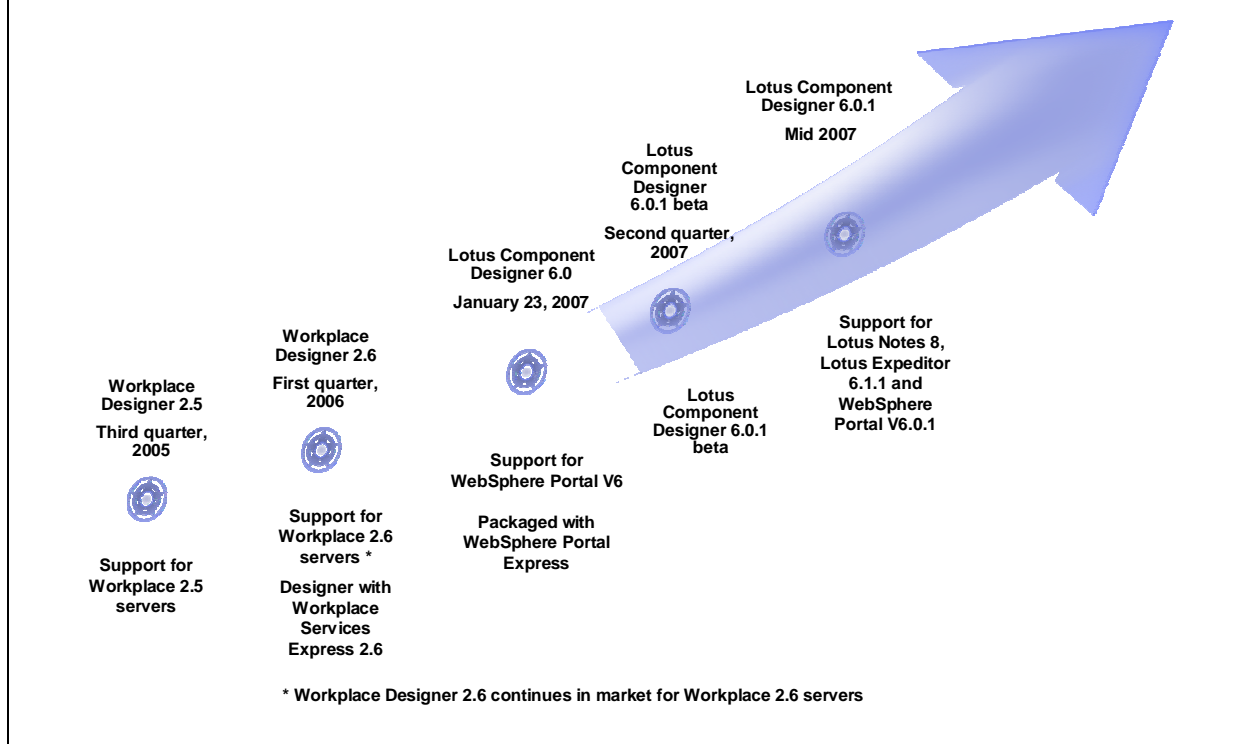
In addition to the statements (shown on this chart) made by Matt Smith, of New York-based E2 Consulting Group, he also says, "IBM has provided exactly what its Business Partners have been hoping for and we expect to use Lotus Component Designer extensively with our clients."

Note: E2 provides high-value consulting to enable the evolution of an IT organization. E2 compliments this consulting with the implementation of emerging technologies to enable its clients to collaborate, improve productivity, lower technology costs and deliver high-impact solutions by using best practices and leading technologies.

Alex el Homsy, who also has used the Lotus Component Designer beta release, works for Trilog Group, a 10-year-old IBM Premier Business Partner and Lotus solution provider that is headquartered in Woburn, Massachusetts (additional offices in France and Lebanon). In addition to the statements (shown on this chart), el Homsy says, "The learning curve was minimal. Our developers actually found Lotus Component Designer more capable and no less agile than Lotus Domino Designer — a very impressive new tool."

Note: Trilog Group has received numerous awards, including the coveted IBM PartnerWorld Beacon Award for Best Lotus Software Solution of 2006 for its flagship ProjExec software product. ProjExec was also honored with the 12th Annual Lotus Award for Best Contextual Collaboration Solution. Trilog Group was presented with the 2006 CTO Innovation Award recognizing visionary partners who are leading the charge in exploiting the latest Lotus collaborative capabilities in innovative, leading-edge and exciting new ways. Also, JWT, a longstanding Trilog Group customer, was awarded the 2006 High Performance Workplace Excellence Award for Business Innovation by Gartner for its project-collaboration portal built by Trilog Group.

Lotus Component Designer road map



Lotus Component Designer road map

This chart shows a continuum of the development investment and forward momentum that IBM is providing for Lotus Component Designer. Lotus Component Designer 6.0 became available in January 2007. This release supports WebSphere Portal V6.0 and WebSphere Portal Express V6.0. It offers an extensible control set, including built-in Ajax controls. It integrates with external data sources and Web services.

An evaluation version of this offering is available for download:

ibm.com/software

Then navigate as follows:

Lotus software → search on **Lotus Component Designer**

You can also find samples and demonstrations at the following Web site:

ibm.com/developerworks/lotus

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

Lotus Component Designer 6.0.1 is available in the summer of 2007 and supports the IBM Lotus Notes® 8 platform, as well as IBM Lotus Expeditor and IBM Lotus Sametime. It will allow you to create native Eclipse components as well as WebSphere Portal components — this function will be included with Domino Designer 8. Component Designer 8 will also support offline synchronization of data.

Note: In the future, Lotus Component Designer will merge with Lotus Domino Designer to become a single Lotus Domino design suite.

Adding support for Lotus Notes and Lotus Expeditor

- **Design one time, deploy to WebSphere Portal and Lotus Notes 8 or Lotus Expeditor 6.1.1**
 - Client applications take advantage of platform
 - Use composite application editor to add components to client-based composite applications
- **Deploy to WebSphere Portal, which provisions to client**
 - Applications provision to client automatically
 - Initial provision creates local XML document data store for you
- **Applications can run disconnected**
 - Synchronization of XML document data between local and server data store are built into the product
- **Full preview functionality**
 - Temporary local deployment to client (live data and execution logic)

Adding support for Lotus Notes and Lotus Expeditor

IBM is enhancing Lotus Component Designer for its 6.0.1 release (in mid-2007). This release will make it possible to design one time and then deploy to both WebSphere Portal and Lotus Notes 8 or Lotus Expeditor 6.1.1. Additionally, client applications will be able to take advantage of the Lotus Notes platform. It will also be possible to use a composite-application editor to add components to client-based composite applications.

Additional support in Lotus Component Designer 6.0.1 will include the ability to deploy portlets to WebSphere Portal that provisions applications to the client. An initial provision creates a local XML-document data store for use by your application.

Applications will also be able to run while disconnected from the Lotus Domino server. In these cases, synchronization of XML document data (between the local-data store and the server-data store) will be built into this next release of the Lotus Component Designer product.

The next release will also support full capabilities to preview newly created or modified Lotus component-based portlets. You will be able to do this by temporarily deploying the portlet locally to the client — you will see live data and execution logic when previewing the code.

Agenda

- **Position development tooling**
 - Rational and WebSphere Studio tools
 - WebSphere Portlet Factory and WebSphere Dashboard Framework
 - Lotus Component Designer
 - ✓ **Lotus Expeditor**

Lotus Expeditor

This final section of the course focuses on IBM Lotus Expeditor (formerly IBM WebSphere Everyplace Deployment).

Lotus Expeditor

Client platform for portal, forms, Sametime, Notes 8 and WebSphere

WebSphere Portal

Single access point for content and applications

- Local portlet for offline and mobile use
 - Compelling, responsive, local user interface
- Composite desktop applications*

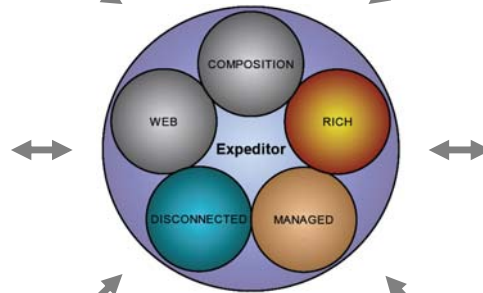
IBM Workplace Forms (not supported on i5/OS)

Forms-based computing to accelerate business

- Forms integrated into composite desktop
 - Offline forms access
- Electronic forms solution for people on the move*

Start with Eclipse

- Open source and standards
 - Rich graphical user interface
 - Extensible component framework
 - Software lifecycle management
- Platform for rich-client applications*



Lotus Sametime

Platform for real-time business

- Links SOA to Sametime
 - Sametime offline
- Messaging in composite applications*

WebSphere Application Server

Software for on demand businesses

- Extending applications beyond the data center
 - Symmetric programming model
- An open alternative to .NET*

Lotus Notes 8

Taking collaboration to a new level

- Composite desktop applications
 - Links people to processes
- Extends SOA to Lotus Notes*

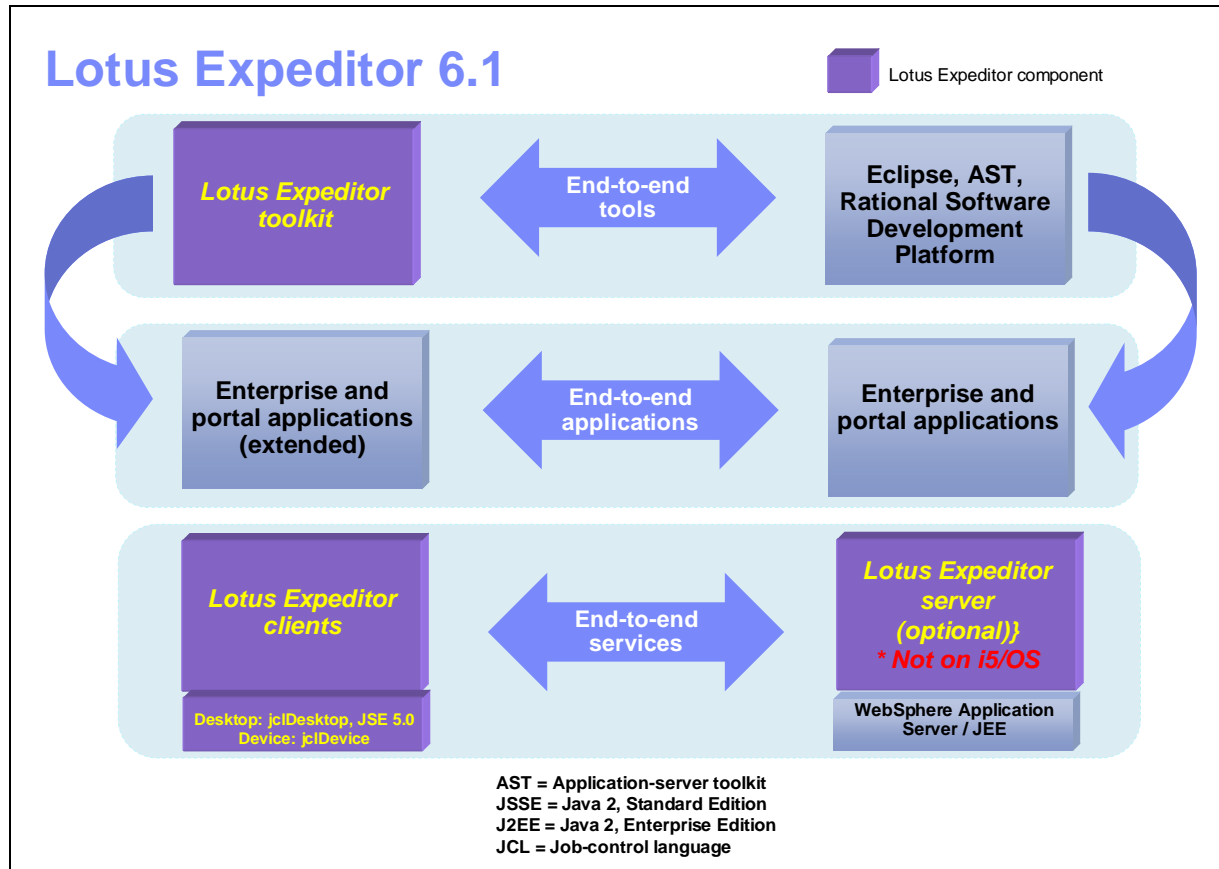
Lotus Expeditor

Lotus Expeditor is the core technology for implementing Lotus rich clients.

Lotus Expeditor is built on Eclipse, which provides the Java technology-based platform for rich-client applications. Eclipse is an open-development community that provides open source and standards for a rich-client platform that runs on a variety of clients, including desktops, laptops, tablets, kiosks and handheld devices. The Eclipse rich-client platform (RCP) supports the development of applications with a rich GUI. Eclipse provides an extensible component framework that supports customers and partners in easily adding services to the client platform. Eclipse also manages the lifecycle of software components that run on the client platform (including the installation, update and removal of components). Multiple components run on a single Java virtual machine (JVM) instance. As a result, multiple applications and services can run on a single JVM without the overhead of launching a JVM to run each application and service.

Lotus Expeditor extends Eclipse to provide a platform for composite applications. Lotus Expeditor provides access services, such as data store, messaging and Web services, that allow composite applications to access information in the IT data center. Lotus Expeditor also supports management of the client platform and applications as well as role-based provisioning. Finally, IBM provides service and support for all components in Lotus Expeditor, including Eclipse.

In addition to being a stand-alone product, Lotus Expeditor also delivers the core set of client services for Lotus Sametime and the next version of Lotus Notes, called Lotus Notes 8.0. IBM Lotus Sametime® software adds the services necessary to provide the platform for instant messaging and conferencing; Lotus Notes adds the services necessary for business e-mail and collaboration.



Lotus Expeditor 6.1

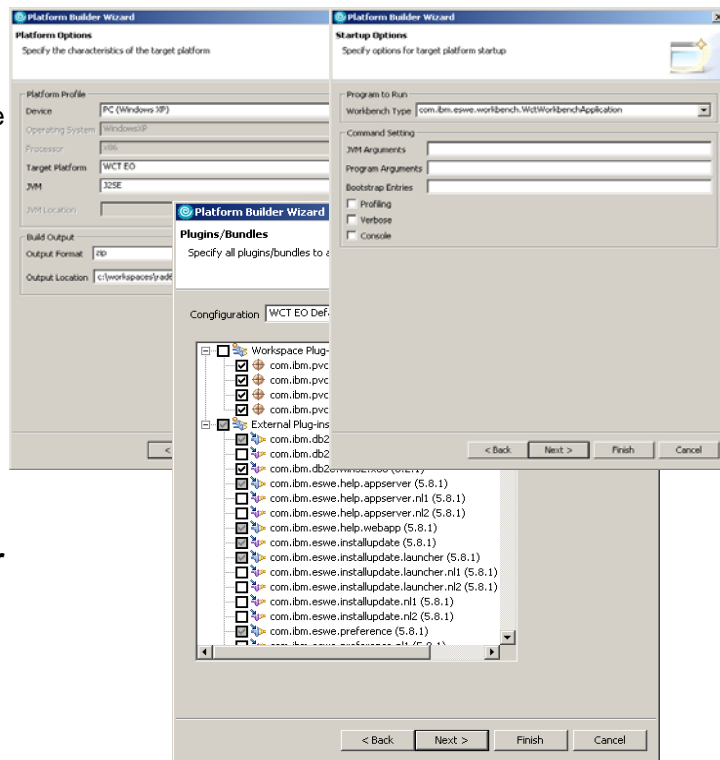
The combination of the Lotus Expeditor client and the Lotus Expeditor server provide the end-to-end services necessary to deliver and manage end-to-end applications. Administrators use the Lotus Expeditor 6.1 server to install and configure the services necessary for client applications to securely perform assured transactions and database synchronization with Enterprise applications and data. Furthermore, the server provides an enterprise-caliber management solution so that a systems administrator can manage client applications and resources. (The Lotus Expeditor server is optional.)

The Lotus Expeditor toolkit provides a complete, integrated set of tools that allows you to develop, debug, test, package and deploy client applications. This toolkit is built on Eclipse technology and extends Eclipse and the powerful Rational suite of development tools so that you can leverage your existing skills and software components to develop end-to-end applications.

Lotus Expeditor toolkit

Tools that plug into Eclipse to build, test and deploy Expeditor applications

- **Targets Java and Eclipse development**
 - Extend IT applications to the desktop
 - Modify for offline and local use
 - Integrate native applications
- **Supports platform APIs**
 - SWT, JSP, portlets, OSGi and others
- **Supports desktop customization**
 - Can be transparent to users
 - Control over desktop
- **Provides a platform builder**
 - Construct fit for purpose run times
 - Minimize footprint



Lotus Expeditor toolkit

You can use the Lotus Expeditor toolkit for rapid application development and deployment of managed-client applications. This product is targeted for developers who want to build portlets and components that interface and work within the Java and Eclipse development environment. You can use this toolkit to extend IT applications to the desktop. You can modify components for offline and local use and you can integrate components with native applications.

The Lotus Expeditor toolkit supports platform-specific APIs, including: SWT, JSP, portlets, OSGi and others.

It is also possible to use the toolkit for desktop-customization projects. It can make the final application's various technologies transparent to the user and provides you with functions to take control of the desktop assets.

A platform builder supports your need to construct applications that are fit for the intended run times, all while running in a smaller footprint.

Summary

- **Positioned following IBM development tooling:**
 - Rational and WebSphere Studio tools
 - WebSphere Development Studio Client
 - WebSphere Portal toolkit
 - WebSphere Portlet Factory and Workplace Dashboard Framework
 - Lotus Component Designer
 - Lotus Expeditor

Summary

As you can see, many development options are available for designing and developing a portal user interface and portlets. Each of these options meets particular needs and can accommodate all development skill sets. Whether you have no development skills or if you are a portal- and portlet-development guru — there is a tool that is right for you and your development organization.

The factors that dictate what option is best for your particular situation include:

1. Who will be performing the creation or development?
2. What is their particular skill set? You likely would not use the Rational and WebSphere Studio tools with no Java development skills. You would instead consider using a code-generation tool such as WebSphere Portlet Factory.
3. Previous tooling experience must also be considered. For example, developers with a Lotus or .NET scripting background can easily adopt the use of the Lotus Component Designer tool.
4. How integrated must your application be with WebSphere Portal? For example, a developer can build tighter integration with WebSphere Portal if by performing more of the overall development, as compared to using code-generation tooling.
5. What is the level of complexity for your requirements? Traditionally, features and functions that are complex or are specific to your business purposes require that you have full control of the development framework. In these cases, code-generation tooling is not recommended.

Check out the product sites for each of these tooling options to learn more about how they fit into your development organization.

Links

- 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
<http://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
- Host Applications Transformation Server (HATS)
ibm.com/software/webrowsers/hats/
- WebFacing Deployment Tool with HATS Technology
ibm.com/software/awdtools/wdht
- WebSphere Portal Toolkit
ibm.com/software/info1/websphere/index.jsp?tab=products/portaltoolkit
- iSeries Access for Web
ibm.com/servers/eserver/series/access/Web
- iSeries Access for Web portlets
ibm.com/servers/eserver/series/access/web/portlets.html
- Lotus Component Designer
ibm.com/lotus/componentdesigner
- Developer resources for Lotus Component Designer
ibm.com/developerworks/workplace/products/designer
- IBM presentation on WebSphere Dashboards
ibm.com/be/pdf/en/events/symposium/portal_v6_and_dashboard.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.

DB2, Domino, eServer, i5/OS, IBM, the IBM logo, Integrated Language Environment, iSeries, Lotus, Lotus Notes, OS/400, RACF, Rational, Redbooks, System i, Tivoli, WebSphere and Workplace 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.

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

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

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

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.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function, or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

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.