



Lotus Expeditor 6.1 Education

Lotus® Expeditor Toolkit Overview

Lotus software



@business on demand software

© 2006 IBM Corporation

Hello and welcome to this overview of Lotus Expeditor Toolkit.

Agenda

- Introduction
- Installation and Configuration
- New Functions
- Detailed Descriptions
- Questions

The goal of this presentation is to introduce the Lotus Expeditor Toolkit. There will be an Introduction to the components that are part of the toolkit, followed by information on how to install and configure it. We'll then talk about the new functions that have been added to the toolkit in this release and provide some details on each.

Overview

- ▶ Bundle Developer Kit (BDK) extends Eclipse PDE to develop client services projects and complements the Eclipse Visual Editor for Java™ to create rich client applications (SWT)
 - PDE - Eclipse Plug-in Development Environment
- ▶ Web Tools work with Rational® Web tools or Eclipse Web Tools Platform (WTP) to deploy Web applications to the client
- ▶ Embedded Transaction Container Tools work with Rational EJB tools to deploy Embedded Transaction Applications to the client
- ▶ Mobile Web Services Tools enable client applications to consume and provide Web Services
- ▶ Portlet Tools work with Rational Portal tools to create JSR 168 portlet applications to deploy to the client
- ▶ Platform Builder enables custom desktop client platforms to be built
 - Check your licensing agreement before you deploy custom client platforms

The toolkit provides a complete, integrated set of tools that allow you to develop, test, debug, package and deploy client applications. You can use the toolkit to develop a variety of applications, including rich client applications, web applications, embedded transaction applications, portlet applications, and web services applications. Within the toolkit there are several tools components that support these activities.

The first one listed here is the Bundle Developer Kit. It provides a framework for creating projects that run on the client platforms that we call Client Services projects and it is an extension of the Eclipse Plug-In Development Environment (PDE).

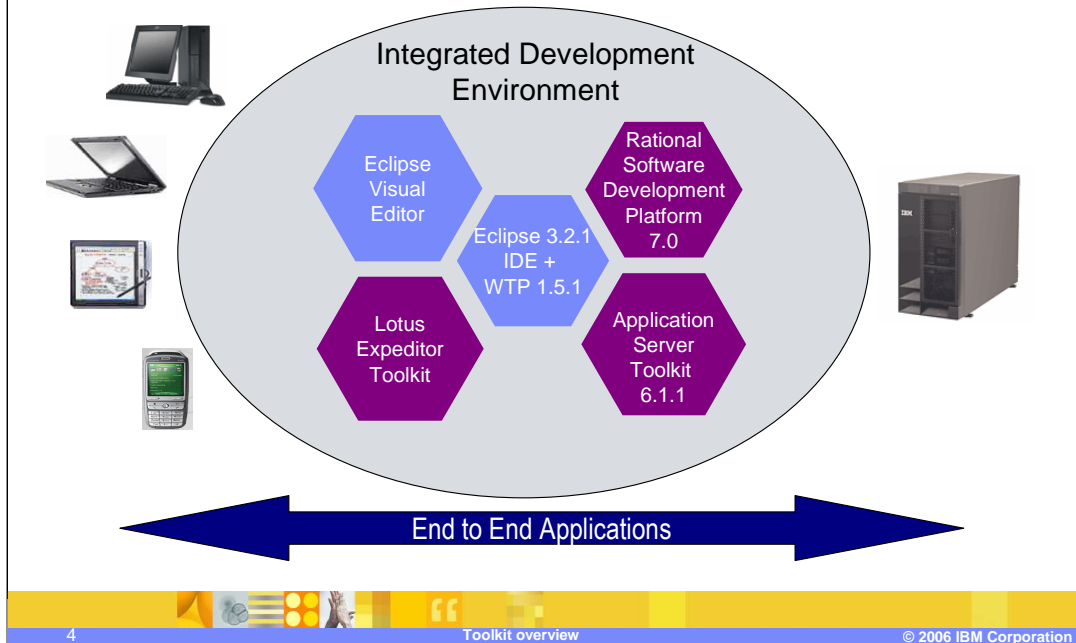
Another major component within the toolkit is the Web Tools that support development of web applications. The Web Tools work in conjunction with either Rational Web tools or within an Eclipse and Web Tools Project environment.

The toolkit also provides Embedded Transaction Container Tools that work with the Rational EJB tools to enable development of Embedded Transaction Applications for the client platforms.

Mobile Web Services tools are provided in the toolkit, which supports development of applications that consume or provide Web Services in a secure manner.

In this release, one of the major enhancements that is being provided are the **Page 8 of 29**
 These tools can be used with the Rational Portal tools to create JSR 168 portlet applications for the client platform

Tools



4

Toolkit overview

© 2006 IBM Corporation

End-to-end tools are available that enable you to develop a variety of client and server applications in a single integrated development environment that is built on Eclipse 3.2.1. Other tools such as the Eclipse Visual Editor can be used to develop rich client platform applications with SWT.

IBM provides tools such as Rational Application Developer (RAD) and Rational Software Architect (RSA) that can be used to develop Enterprise server applications. These products provide tools to develop Web applications, Web Services, Portlets and EJB's. In addition to these development tools, Rational Software Architect provides integrated design and development tools that leverages model driven development. During this presentation, you may see the term Rational Software Development Platform used. This is a general term that refers to either RAD or RSA.

Alternatives to RAD and RSA, including the IBM WebSphere® Application Server Toolkit (AST) or Eclipse 3.2.1 with the Web Tools Project are also supported. These platforms are not functionally equivalent to RAD and RSA, therefore users will not be able to take advantage of the full set of toolkit capabilities within these environments.

Lotus Expeditor 6.1 Toolkit – Software platforms

- **Windows® XP or RedHat EL 4.0 WS**
 - ▶ Rational Software Architect V7.0, Rational Application Developer V7.0 or WebSphere Application Server Toolkit 6.1.1
 - Samples Gallery and WS-Security editors
 - Full Java Server Faces (JSF) support with Rational Application Developer or Rational Software Architect
 - Supports development of Client Services projects, Web applications, Web Services, Embedded Transaction applications, Portlets and Platform Builder projects
 - ▶ Eclipse 3.2.1 and Web Tools Project (WTP) 1.5.1
 - Supports development of client services projects, Web applications, Web Services and Platform Builder projects

The Lotus Expeditor Toolkit can be installed on RSA, RAD, AST or the Eclipse 3.2.1 IDE. With RSA or RAD, you will have access to the complete set of toolkit functions, including the Samples Gallery, WS-Security editors and full JSF support. In addition, you will be able to develop client services applications, Web applications, Web Services, Embedded Transaction applications, Portlets and Platform Builder projects. With AST, you will have access to the same set of capabilities provided in a RSA or RAD environment, with the exception of JSF support.

The Eclipse 3.2.1 + WTP 1.5.1 environment is intended for those customers who need to develop Client Services projects, Web applications, Web Services and Platform Builder projects.

If you are developing for devices, note that the Portlet and Platform Builder components are not available.

Lotus Expeditor 6.1 Toolkit - Prerequisites

■ Hardware Requirements

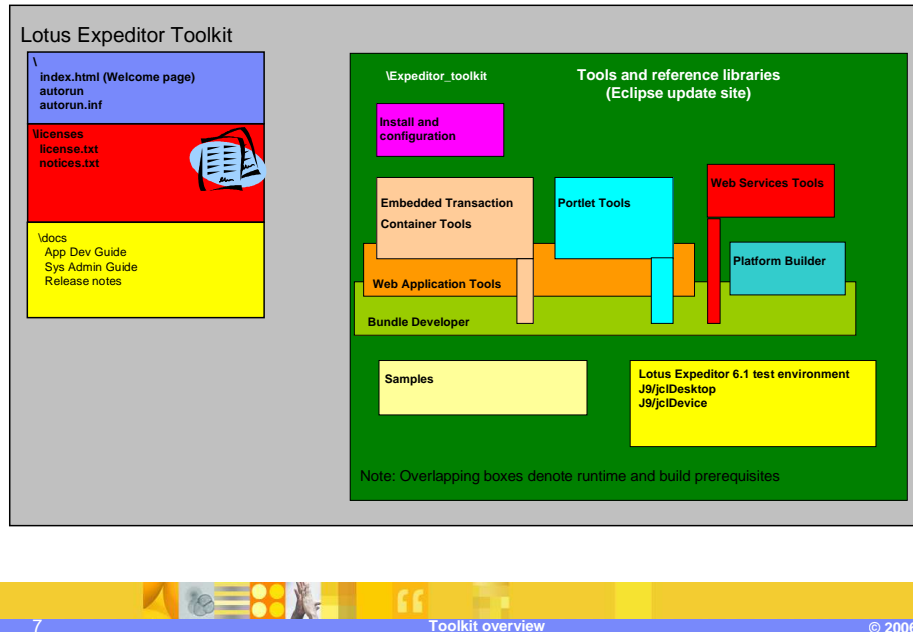
- ▶ 512 MB RAM
- ▶ 500 MB free disk space
- ▶ x86 processors capable of supporting Windows XP/SP2 or RedHat EL 4.0 WS with GTK Support – Update 3
- ▶ Super vide graphics array/adaptor (SVGA) (800 x 600) display minimum (1024 x 768 recommended)

■ Software Requirements

- ▶ Microsoft Windows XP Service Pack 2 or RedHat EL 4.0 WS with GTK Support – Update 3
 - IBM Rational Software Architect V7.0, IBM Rational Application Developer V7.0, WebSphere Application Server Toolkit 6.1.1 or Eclipse 3.2.1 with Web Tools Project 1.5.1

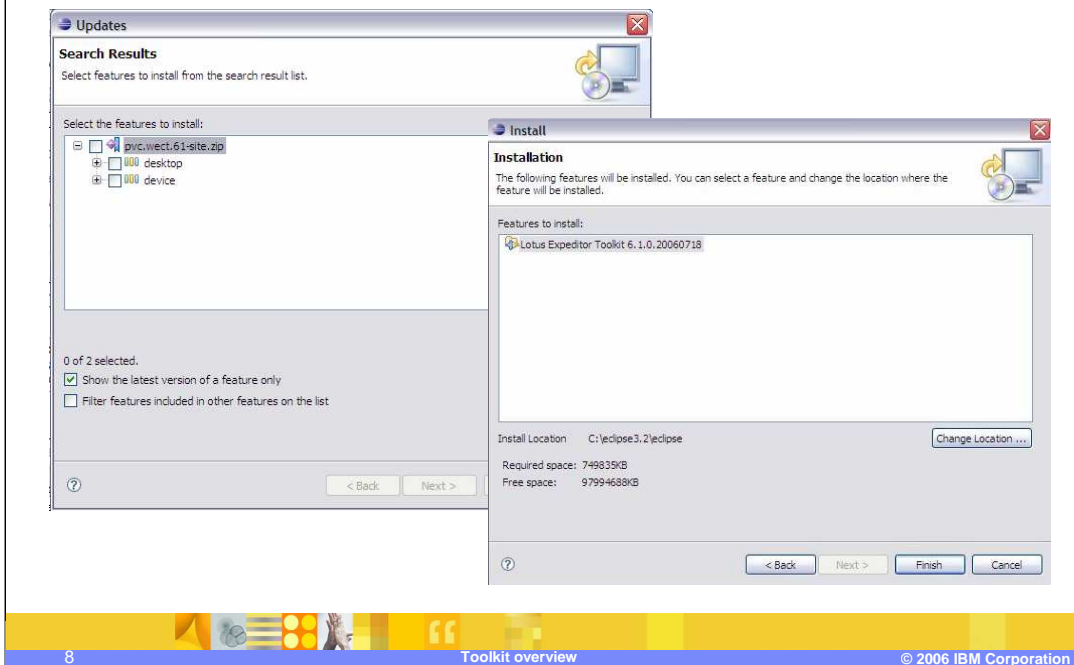
You must ensure that each machine on which you plan to install the IBM Lotus Expeditor 6.1 Toolkit meets the requirements shown on this slide.

Toolkit packaging



At this point in the presentation, we'll move into discussion of the second item on the agenda, which is toolkit installation and configuration. When the toolkit package is downloaded from the website, the contents will be similar to what is depicted here. There will be a licenses and documentation directory in the package as well as a toolkit directory where the tools components and JREs will be packaged.

Installation steps



The toolkit is installed as an Eclipse update. The desktop toolkit installation installs as a single feature for ease of installation. The device tools are optional and can be installed when the desktop toolkit is installed or later on as a separate update.

To install the toolkit, select Help > Software Updates > Find and Install in the IDE. For a first time install, you'll select Search for new features to install and then press Next.

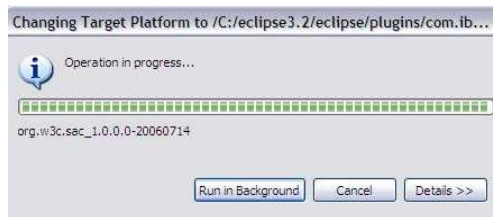
Next, add the toolkit as an update site and press Finish. This will take you to a dialog that enables you to select the features to install. Select desktop for the desktop toolkit install and optionally select device to add device tools. You will then be prompted to accept the terms of the license agreement, and following that a dialog will appear that shows you a summary of the features that will be installed. Press Finish on that dialog to perform the installation. You will receive a prompt as the desktop toolkit feature and the device feature are installed. Select Install on each of those dialogs to continue with the installation process. Once the installation process finishes, you will receive a prompt to restart the IDE, which is typical after an Eclipse update is performed.

Configuration steps

- Prompt to set toolkit preferences on first startup or when new workspace is opened.



- Makes jclDesktop the default JRE and sets the Eclipse target platform to enable applications to be developed that target the client platform.

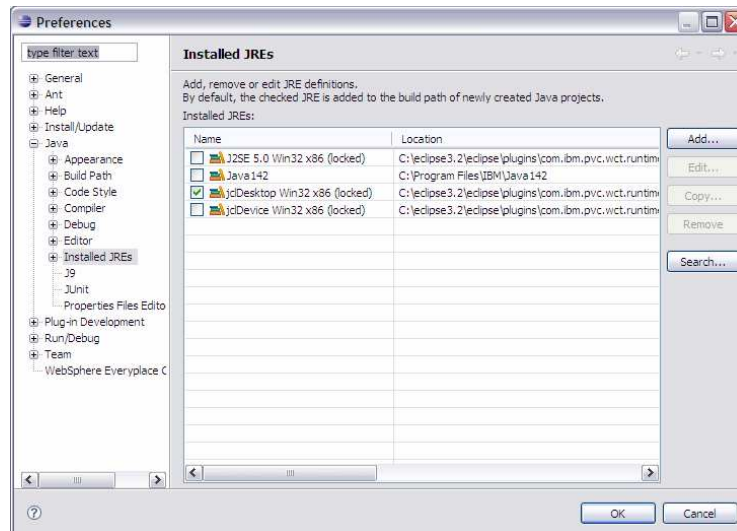


When the IDE starts up for the first time after the toolkit is installed, or when the IDE is opened using a new workspace, you will need to set preferences for the toolkit before you start developing applications for the Lotus Expeditor client platform. Configuration assistance is given in the form of a pop-up dialog on startup. The pop-up will ask if you wish to set these preferences, and by pressing OK in this dialog, the required steps will be performed for you. The preference options that are set consist of setting the default JRE to jclDesktop, setting the compiler compliance level, and setting the Eclipse target platform.

Once the toolkit is installed and configured you will be able to start developing applications that target the Lotus Expeditor client platforms.

Changing JRE selection

- Change the default JRE selection using the Installed JREs Eclipse preferences



You can change the default JRE by setting the standard Eclipse preference option found under Java > Installed JREs. You might do this for example if you are developing for devices, and you need to use the device JRE which is called jcdDevice. When developing applications for the client, you should always select one of the JREs that are contributed when the toolkit is installed. This will ensure that the environment you are using for development and test within the IDE matches the target runtime.

Now that you have an understanding of the toolkit components and how to install and configure it, let's go to the next part of the presentation and discuss the new functions that have been added to the toolkit in this release.

New functions

- Portlet Tools
 - ▶ Build and deploy disconnected portlets running on Lotus Expeditor
 - ▶ Convert JSR-168 portlet projects to disconnected portlets
- Support for new client runtimes including launch, debug, build and export
 - ▶ J9/jclDesktop
 - ▶ J9/jclDevice
- Lotus Expeditor project templates

In this release, the following new functions have been added. One of the major enhancements in the toolkit is the addition of Portlet tools. These tools allow users to create Portlet applications that conform to the JSR 168 standard. These portlets can be run on the desktop client in the portlet container. This container supports running portlet applications either disconnected or connected and in addition to support for developing new applications, the toolkit allows conversion of existing portlet applications into Client Services Portlet applications that will run on the Lotus Expeditor client platform.

Another new function that has been added in this release is support for the latest version of the Lotus Expeditor client runtimes. The toolkit provides an application development environment along with a packaged test environment for users to develop applications that run on the client platform. Users can create, build, debug and export projects that are targeted for use with the client. In addition, applications can be developed using jclDesktop or jclDevice to target the desired version of the client (desktop or device).

In this release, we have also added project templates that can be used when creating Client Services projects. These templates are provided as a means of jump starting application development.

Enhancements

- Installs into RSDP 7.0 (Eclipse 3.2.1-based)
- Web Tools
 - ▶ Improved JSF/JSTL integration
 - ▶ Integration with Eclipse Web Tools Project (WTP)
- Embedded Transaction Tools
 - ▶ Improved validation and developer notification
- Web Services and WS-Security Tools
 - ▶ Supports WS-Resource Framework (WSRF)
 - ▶ Support for Axis 1.3 and JSR-172 clients
- Platform Builder
 - ▶ Provides extensions to Eclipse product configuration editor
 - ▶ J9/jclDesktop and Java SE 5.0 support

In addition to these new functions, there are numerous enhancements that have been made to the toolkit in this release. As mentioned in the Overview, the toolkit is Eclipse 3.2.1 based and this is the version of Eclipse that is included in version 7 of the Rational Software Development Platform.

Next, the Web tools have been modified in this release to extend the Web tools provided by the Eclipse Web Tools Project and in addition, we have improved JSF and JSTL integration. In this release, the JSF and JSTL libraries are no longer copied into the Client Services web projects and therefore these applications can be run without the need to manually remove these libraries.

In the Embedded Transaction Tools, there is improved validation and developer notification. For example, when a non-Client Services project is converted to a Client Services project, the Embedded Transaction Tools will check the project for any EJB functionality that is not supported by the client runtime. If any such functionality is found, a panel is presented to the end user listing the details of the unsupported functionality.

In the Web Services Tools, we have support for Axis 1.3 and JSR-172 clients. In addition, support was added for the Web Services Resource Framework (WSRF).

Finally, in this release, the Platform Builder component has been modified to extend the Eclipse product configuration editor and it now supports export of a platform that includes jclDesktop in addition to Java SE.

Bundle developer kit

- Extends Eclipse PDE tools
- Provides project creation and configuration
- Application templates
 - ▶ Basic Client Services
 - ▶ Rich Basic Application
 - ▶ Preference Page Basic Application
- Automated manifest management
- Provides set of PDE Target definitions
- PDE style launch
 - ▶ Java SE, jclDesktop and jclDevice configurations
 - ▶ Launch using selected Target definition

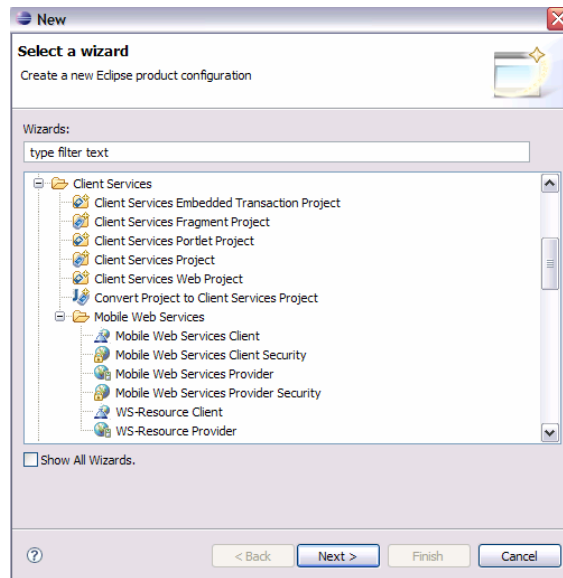
Now let's take a look at the technical details of the new functions and enhancements that we've discussed. The Bundle Developer Kit extends Eclipse's PDE tools and it is the component in the toolkit that is responsible for creation of Client Services projects.

During project creation, the user can select from a list of available Target definitions. Target definitions provide a convenient way to specify a set of runtime components and you can select components to be added to your project's build path when creating the project. In addition, the Bundle Developer Kit can provide automated manifest management. This means that the Require-Bundle and Import-Package entries in the project's manifest file are updated based on changes in the project's Java code. When new package dependencies are added to the project's Java code, the manifest file is automatically updated with the proper entries.

Another option that has been added during Client Services project creation, is selection of an application template. Templates are provided that enable you to create a basic client services project, a basic preference page application, or a rich basic application to get a jump start on creating a rich client application with a perspective and a view.

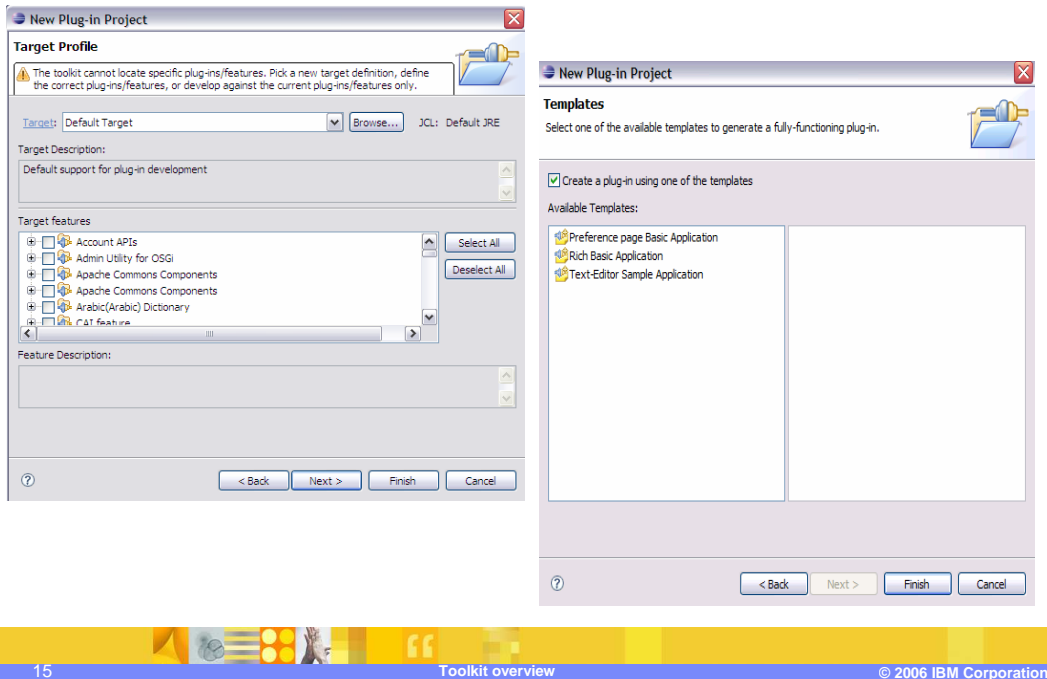
Finally, the Bundle Developer Kit provides a PDE style launch. This can be used to test applications and we have customized the launch configuration to add selection of the Target Profile and JRE.

Client services project types



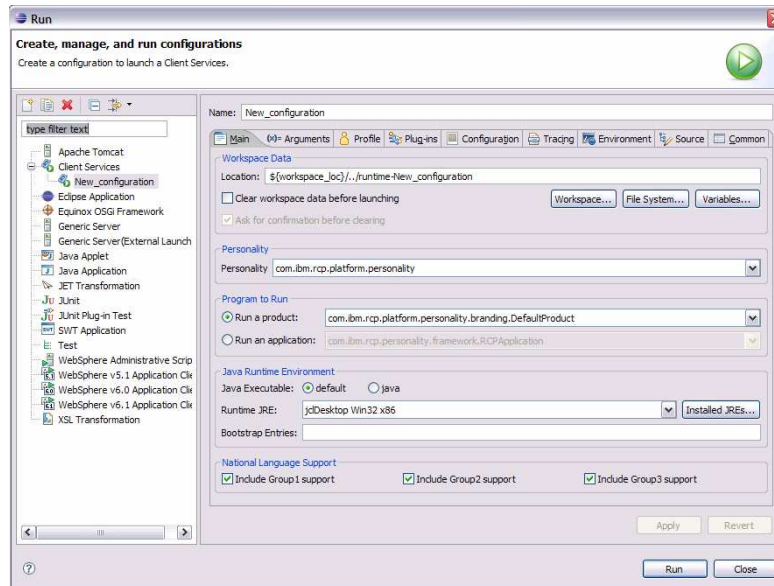
Here, there is a list of the types of Client Services project that are supported in the toolkit. You can see that there are Client Services projects and fragment projects, Embedded Transaction projects, Web Projects, Web Services Projects and Portlet Projects.

Project creation – Client services project



During Client Services project creation, the wizard page on the left allows you to select the Target definition and then the runtime components to add to your project's build path. On the right, there is a screen capture of the project creation wizard page that allows you to select a template. To create a basic client services project, do not highlight the template name and merely click Finish in this dialog.

Launch



On this chart, you can see the Bundle Developer Kit's launch configuration dialog. This is a basic PDE-style launch but we have added a Target tab that enables selection of the Target definition that should be used for the launch.

Target definitions

- Replacement for Platform Profiles
- Provide grouping of features or plug-ins in the client platform
 - ▶ Core
 - ▶ Default
 - ▶ Default Device
- Also defines related product, application, JVM and personality
- Users can add Target definitions

In this release, we are providing a replacement to the Platform Profiles that we had in the last release that are called Target definitions or Targets. This is a concept that is new in Eclipse 3.2, but is similar to the Platform Profile concept we had in our last release. The idea is to provide a means of grouping the features and plug-ins in the client platform. There are three Targets provided. The first is the Core Target. This Target is similar to the Core Platform Profile from the last release which was used with applications that had no GUI. The second is the Default Target that can be used for applications that have a user interface and last the Default Device Target is used when developing applications for devices.

In addition to defining a grouping of features and plug-ins in the client platform, the Targets can define the product, application, JRE and personality that are intended for use with the Target. Users can define their own Targets in addition to those we provide.

Platform builder

- Allows packaging of applications and custom client platform (desktop)
- Eclipse Product Configuration wizard used to start
- Editor provided to manage the configuration
- Editor extends the Eclipse Product Configuration Editor
- Can select operating system of client platform
- Starts with list of features or plug-ins in Target definition
- User can add from the target platform or workspace
- Supports inclusion of language groups and JVM
- Allows test and export of configuration before deployment

The Platform Builder component allows you to package your applications along with features and plug-ins from the client platform. In this release, custom client platforms can be created that target the Lotus Expeditor Client for Desktops offering. The Lotus Expeditor Client for Devices offering is not supported.

In this release, there is no longer a Platform Builder project type. The standard Eclipse Product Configuration wizard is used to create a configuration file within any project in the workspace. After the configuration file is created, an editor is used to manage the configuration. The editor provided as part of Platform Builder is an extension of the Eclipse Product Configuration Editor. The customizations that have been implemented in the editor include selection of a Target definition, selection of a JRE, selection of one or more language groups, and the addition of features and plug-ins from either the Target or the client platform. After making updates in the editor you have the option to test the configuration and to export it into a directory or a .zip file.

Another capability that Platform Builder provides is that it allows you to select the operating system that the custom client platform will be deployed on. The significance of this is that you can package a platform for one operating system (for example Windows) while running the IDE on Linux[®]. Before deploying custom client platforms, you should check your licensing agreement.

Wizard and editor

The screenshot displays two windows from the Lotus Expeditor 6.1 Education software. On the left is the 'New Product Configuration' wizard, and on the right is the 'Overview' editor page.

New Product Configuration Wizard:

- Product Configuration:** Create a new product configuration and initialize its content. The file name must end with ".product".
- Enter or select the parent folder:** A text field contains "test". Below it is a file explorer view showing a folder named "test".
- File name:** A text field contains "pojproduct".
- Initialize the file content:**
 - Create a configuration file with basic settings
 - Use an existing product: com.ibm.esupport.client.Product
 - Use a launch configuration:
- Navigation buttons: < Back, Next >, Finish, Cancel.

Overview Editor:

- Product Definition:** This section describes general information about the product.
- Specify the name that appears in the title bar of the application:** Product Name: Lotus Expeditor Client
- Specify the initial Target configuration for this product:** Target: Default Target (with a Browse... button)
- Specify the product identifier:** Product ID: com.ibm.rcp.platform.personality.branding.DefaultProduct (with a New... button)
- Specify the application to run when launching this product:** Application: (empty dropdown)
- Specify the Target Environment for the application:** Target Environment: PC (Windows XP)
- Specify the Java Virtual Machine to include in the product:** Java Virtual Machine: j2Desktop Win32 x86
- The product configuration is based on:** plug-ins, features
- Navigation tabs: Overview | Configuration | Launcher | Branding

Here, there are screenshots of the product configuration wizard and editor. The wizard is shown on the left, and the editor is on the right. The editor is a multi-page editor with an Overview, a Configuration, a Launcher and a Branding page.

Web services tools

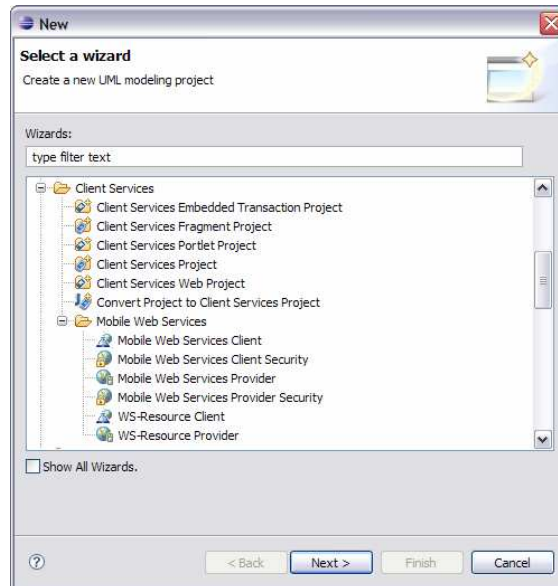
- Support for Web Services similar to JSR-172
 - ▶ Client Stub Generator wizard
 - ▶ Provider Configuration wizard
 - ▶ WS-RF wizard
 - ▶ WS-Security wizard and editors

- Support for Axis 1.3 (JSR-101) Web Services
 - ▶ Client side tools
 - ▶ Integration with Accounts API

Listed here are the capabilities in the toolkit for developing Mobile Web Services and Axis Web Services. For Mobile Web Services, the Web Services tools allow you to generate client code that consumes Web Services and to expose OSGi services as Web Services providers. An application that will consume a Web Service needs to identify the service end-point, typically a URL to a WSDL and then use the interface to invoke the Web Services provider. An application that will be exposed as a Web Services provider must implement a Java interface that defines the Web Service calls. There is a Mobile Web Services Client and a Mobile Web Services Provider wizard within the toolkit to help you accomplish these tasks. Security can be configured using the client and provider wizards. There are also Web Services Security Editors available that allow you to editor your Web Services security configuration for the client and provider. These editors are available in RAD, RSA and the AST environments.

In addition to developing Mobile Web Services, the toolkit can also be used to generate Apache Axis based Web Services client code and the client platform's accounts API can be used to define an Account for the Web Services.

Web services tools



This slide depicts the available Mobile Web Services wizards. The WS Resource Client and Provider wizards are part of the Web Services Resource Framework.

Web application tools

- Construct Web Applications
 - ▶ JSP, Servlet, JSF, JSTL, and Struts support
 - ▶ Deploy and Execute on the client runtime
- Convert or Retarget
 - ▶ Run and debug Expeditor Applications on WebSphere Application Server
 - ▶ Run and debug WebSphere Applications on Expeditor

Web Application tools enable development of Client Services web applications that use JSPs, servlets, Java Server Faces (JSF), JSTL which are JSP standard tag libraries and finally Struts. When creating an application for a device, Struts, JSF and JSTL are not supported.

These Web Applications can be created, tested and debugged within the toolkit environment and can be exported and run on the client runtime like the other project types that the toolkit supports.

There are options available for running and debugging Client Services Web Applications on a WebSphere Application Server and for running and debugging a WebSphere (or dynamic web application) on the Lotus Expeditor client. First, you can convert an existing WebSphere Web project into a Client Services Web project by using the Convert Project to Client Services project wizard. This will add the Lotus Expeditor artifacts to the project and the project's classpath is maintained to match the class path environment that will exist in the runtime.

In addition, a dynamic web application can be run on the Lotus Expeditor client platform by targeting the project's runtime to the Client Services runtime using the project's Targeted Runtimes properties.

Client services Web project

The screenshot displays the Project Explorer on the left, showing the structure of a Client Services Web project. The project contains several folders and files, including:

- Deployment Descriptor: CS Web Project
- Security Editor
- Web Diagram
- Java Resources
 - src (labeled as Java code)
 - WebContent\WEB-INF\classes
- Libraries
- build
- META-INF
- MANIFEST.MF (labeled as Expeditor artifacts)
- WebContent
 - META-INF
 - WEB-INF
 - build.properties
 - plugin.xml (labeled as Expeditor artifacts)
 - Web Site Navigation

Red arrows point from the labels to the corresponding files/folders. Blue arrows point from the labels to the META-INF and MANIFEST.MF files.

In the lower left, a context menu is shown for the 'Run As' option, with 'Run on Server' selected. The keyboard shortcuts for 'Run on Server' (Alt+Shift+X, R) and 'Eclipse Application' (Alt+Shift+X, E) are visible.

On the right, the 'Run On Server' wizard is shown. The 'Define a New Server' step is active. The wizard prompts the user to choose the type of server to create. The 'Manually define a new server' option is selected. The server's host name is set to 'localhost'. The server type is set to 'Client Services v6.1'. The description is 'Client Services v6.1'. The server runtime is set to 'Client Services v6.1'. The 'Set server as project default' checkbox is unchecked.

Here, in the upper left, we show the plugin.xml and MANIFEST file that are present in a Client Services Web project. These are Lotus Expeditor artifacts. Also shown is the location of Java code and web content. In the lower left and on the right there are screen captures of the Run on Server context menu and wizard that are used to run the Web application. When you want to test an application that is targeted for Lotus Expeditor within the toolkit environment, you should select the Client Services server type.

Portlet tools

- **Construct JSR168 compliant portlet applications**
 - ▶ Faces, Basic and Empty portlet type support
 - ▶ Deploy and Execute on client runtime
- **Convert or Retarget**
 - ▶ Run and Debug WebSphere Applications on Expeditor
 - ▶ Run and Debug Expeditor Applications on WebSphere
- **Limited Portal type functional support**
 - ▶ Aggregation of portlets in JSP is supported
 - ▶ Portlet communication by way of wiring is supported

Again, the Portlet tools represent new function in this release. These tools allow JSR 168 compliant portlet applications to be constructed. When creating a Client Services portlet project, you can select the portlet type from a drop down menu in the wizard. Empty, Basic and Faces portlet types are supported. The portlet development tools within the toolkit allows users to create, develop, test and debug Client Services portlet applications.

In addition, the toolkit provides the option of converting an existing WebSphere Application Portal Server based JSR 168 portlet application into a Client Services portlet project. And, you can also run and debug WebSphere JSR 168 portlet projects on Lotus Expeditor using the same Run on Server ability that is used when testing Web applications.

Finally, there is limited support for aggregating portlets in JSPs and for wiring Client Services portlets. Wiring is used to facilitate inter-portlet communication, and aggregation allows multiple portlets to be displayed on a web page.

Portlet project

The screenshot displays two windows from an IDE. The left window, titled 'New Portlet Project', is a wizard for creating a portlet project. It includes the following fields and options:

- Project name:** CSP
- Project contents:** Use default
- Directory:** C:\IBM\RSA_20060814\workspaces\bw1\CSP
- Target Runtime:** Client Services v6.1
- EAR Membership:** Add project to an EAR. EAR Project Name: EAR
- Portlet API:** JSR 168 Portlet (Standard portlet API according to the Java Portlet Specification Version 1.0 (JSR 168)).
- Create a portlet:** Create a portlet. Portlet name: CSP. Portlet type: Faces portlet (Create a portlet that uses the JavaServer Faces Portlet Framework (JSR 168)).
- Show advanced settings

The right window, titled 'Project Explorer', shows the project structure for 'CSPP':

- CSPP
 - Deployment Descriptor: CSPP
 - Web Diagram
 - Java Resources
 - build
 - META-INF
 - Portlet Deployment Descriptor
 - WebContent
 - _CSPP
 - jsp
 - html
 - CSPPPortletView.jsp
 - META-INF
 - WEB-INF
 - build.properties
 - plugin.xml
 - Web Site Navigation

25

Toolkit overview

© 2006 IBM Corporation

This slide depicts the project creation wizard for a Client Services portlet project and the project layout of the project. Remember that portlet tools are not available if you are using Eclipse and the Web Tools Project as your IDE.

Embedded transaction tools

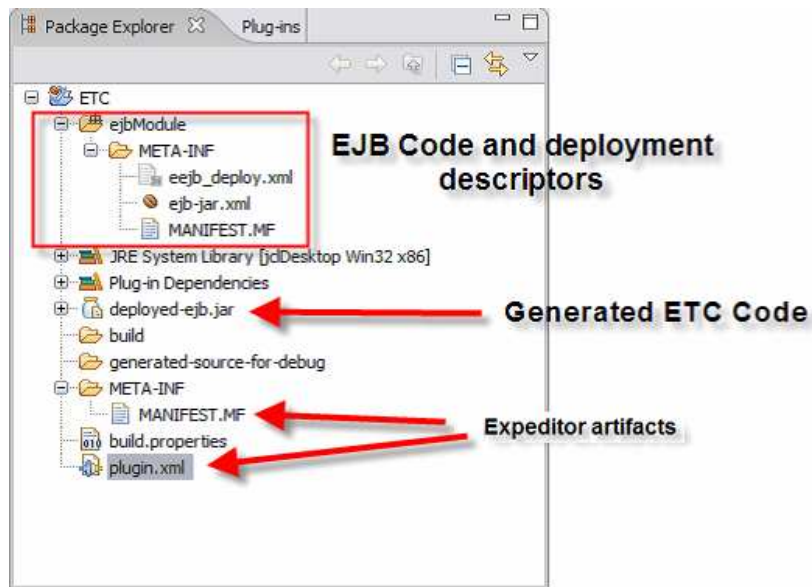
- ▶ The runtime supports a subset of EJBs
 - Stateless Session Beans
 - Entity Beans (BMP and CMP)
- ▶ Construct runtime compliant EJBs
 - Creating new Client Services Projects
 - Converting existing Java EE EJB Projects
- ▶ Usage of functionality not supported by runtime reported
 - During conversion
 - Within project in workspace
- ▶ Deploy, run and debug on client runtime
- ▶ Enhanced validators and quick fixes

The Embedded Transaction Tools are designed to support the subset of EJBs that are supported in the runtime. There is support for stateless session beans as well as Entity Beans, both bean managed persistence (BMP) and container managed persistence (CMP).

Consistent with the other tools components, the toolkit supports both creation of new Client Services Embedded Transaction projects as well as conversion of an existing EJB project to a Client Services Embedded Transaction project. It also provides for test and debug of Client Services Embedded Transaction projects, and for exporting them to be deployed to the client platform.

When developing or converting a project, we have added the capability in this release to detect functionality that is not supported by the runtime and to report errors. There are also project validators and some quick fixes provided to assist in error correction.

Embedded transaction project



This slide gives an example of the contents of an Embedded Transaction Container project. We have the same Lotus Expeditor artifacts here that we did with the other Client Services project types, and you can see the location where the source and generated code reside in the project.

Migration

- ▶ Projects from a previous toolkit release will be migrated when:
 - A workspace that contains projects created using the previous release is opened
 - A project created using the previous release is imported into the workspace

- ▶ Version 5.7.1 and 6.0 projects are migrated for most components

- ▶ There are limitations such as:
 - Web applications and Embedded Transaction Container applications created using RAD or RSA are not migrated in an Eclipse 3.2.1 or AST environment
 - Platform Builder does not migrate 5.7.1 projects

The Lotus Expeditor Toolkit will migrate projects that were created using previous toolkit versions for use with the V6.1 tools and client platforms. After the Lotus Expeditor Toolkit is installed, if you open a workspace that contains projects created using the tools from a previous release, the projects contained in that workspace will be migrated. While convenient, this approach is not recommended since once migration occurs, you will no longer be able to use the workspace with the previous release of the tools.

Projects can also be migrated by importing the project from a previous workspace into a workspace that has been configured for Lotus Expeditor 6.1. When you do this, you will have the option to copy the project into the workspace, which allows you to preserve it in its original form.

With the exception of Platform Builder, the Lotus Expeditor V6.1 tools components migrate projects from the 5.7.1 and 6.0 releases. Platform Builder migrates only projects from the 6.0 release since it does not have support devices in this release. In addition, Web applications and Embedded Transaction Container projects created with Rational Application Developer or Rational Software Architect will not be migrated in an Eclipse 3.2.1 or AST environment.

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 Lotus Rational WebSphere

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

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

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

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

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 or changes in the products or programs 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 (for example, 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 2006. 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.

This concludes this presentation.