



IBM Software Group | Lotus Expeditor 6.1.1 Education

# IBM® Lotus® Expeditor 6.1.1

## Toolkit overview

**Lotus** software



@business on demand software

© 2007 IBM Corporation

This presentation provides an overview of IBM Lotus Expeditor version 6.1.1 Toolkit

## Training objectives

At the conclusion of this training, the audience should ...

- Understand what types of client applications can be developed with the Lotus Expeditor Toolkit
- Comprehend how to develop, debug, and test client applications with the toolkit
- Know which tools are necessary to assemble applications for the Lotus Expeditor client

## Audience

This training presumes the audience ...

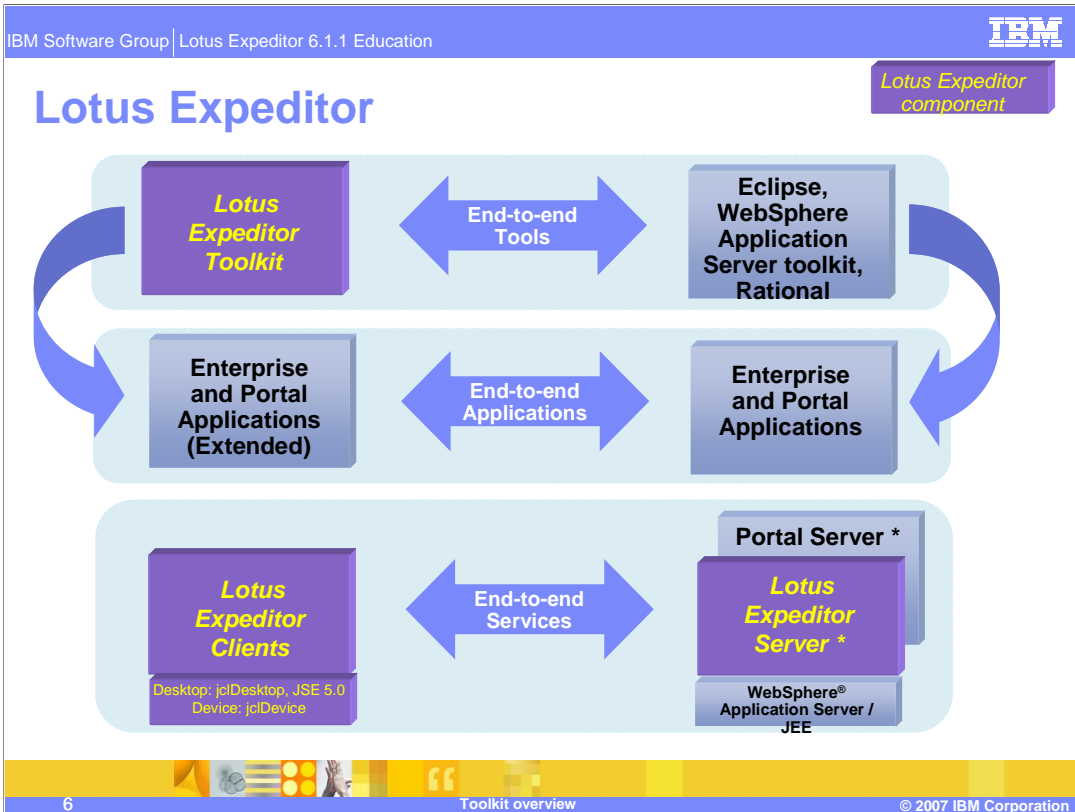
- Knows how to program in Java™
- Understands Eclipse
- Comprehends Rational Software Development Platform

## Agenda

- Introduction
- System requirements for the toolkit
- Installing and configuring the toolkit
- Using the Lotus Expeditor Toolkit with Lotus Expeditor
- Using the Lotus Expeditor Toolkit with Lotus Sametime
- Using the Lotus Expeditor Toolkit with Lotus Notes 8
- Using the toolkit for packaging and deployment
- References

## Tool to develop client applications for Lotus Expeditor

- Lotus Expeditor Toolkit
  - ▶ Develop client applications for Lotus Expeditor
  - ▶ Convert existing projects to client applications
  - ▶ Develop client applications for Sametime 7.5+
  - ▶ Develop client applications for Notes 8
  - ▶ Build custom client platforms - requires an OEM license
  - ▶ Extends familiar Eclipse and Rational tools to target applications at managed clients
  - ▶ Use samples to jump start development



The Lotus Expeditor product consists of a client, server, and toolkit.

The combination of the Lotus Expeditor clients 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 a systems administrator can manage client applications and resources. Note that the 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 you can leverage your existing skills and software components to develop end-to-end applications.

## Use the Lotus Expeditor Toolkit to develop powerful client applications

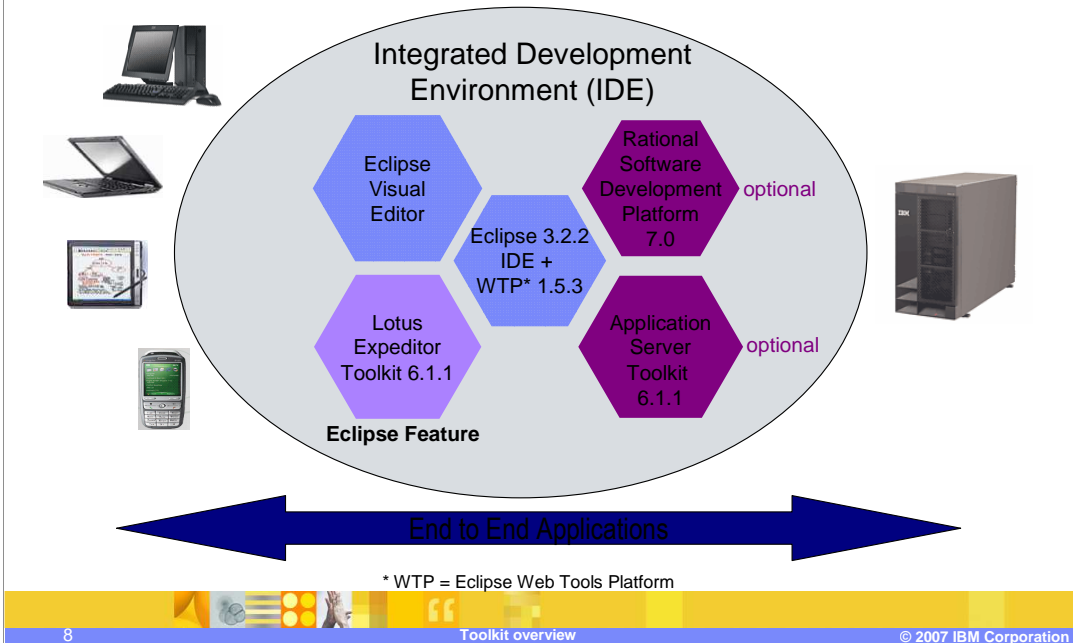
- Develop these types of client applications:
  - ▶ Eclipse rich client platform applications (desktop client only)
  - ▶ Eclipse embedded rich client platform applications
  - ▶ Web applications
  - ▶ Embedded transaction applications
  - ▶ Portlet applications (desktop client only)
  - ▶ Database applications
  - ▶ Messaging applications
  - ▶ Web services applications
- Utilize familiar programming models, API's, and tools to develop these applications

The Lotus Expeditor Toolkit provides a complete, integrated set of tools that allows you to develop, debug, test, package and deploy client applications that use Client Services. You can use the toolkit to develop these types of client applications:

- Eclipse Rich Client Platform applications (desktop client only)
- Eclipse embedded Rich Client Platform applications
- Web applications
- Embedded Transaction applications
- Portlet applications (desktop client only)
- Database applications
- Messaging applications
- Web Services applications

Developers utilize familiar programming models, APIs, and tools to develop these applications.

## Software development platform



IBM's Software Development Platform is built on Eclipse. The Lotus Expeditor Toolkit is installed as an Eclipse feature into Eclipse, Rational, or the Application Server Toolkit IDEs. The toolkit extends existing tools in Eclipse, Rational, and WebSphere Application Server toolkit. The types of applications you can develop with the Lotus Expeditor Toolkit depends on which IDE you install the Lotus Expeditor Toolkit in to. Optionally, you can use the Eclipse Visual Editor to develop rich client applications with SWT.



## Agenda

- Introduction
- **System requirements for the toolkit**
- Installing and configuring the toolkit
- Using the Lotus Expeditor Toolkit with Lotus Expeditor
- Using the Lotus Expeditor Toolkit with Lotus Sametime
- Using the Lotus Expeditor Toolkit with Lotus Notes 8
- Using the toolkit for packaging and deployment
- References

## Operating systems supported

- Lotus Expeditor Toolkit runs on these operating systems and supports application development for the Lotus Expeditor runtime on these same systems:
  - ▶ Microsoft® Windows® XP Service Pack 2
  - ▶ Microsoft Windows Vista™
  - ▶ RedHat Enterprise Linux® 4.0 WS with GTK support – Update 4
  - ▶ SUSE Linux Enterprise Desktop, version 10 (SLED 10)
  - ▶ RedHat Enterprise Linux 5.0 WS with GTK Support – Eclipse 3.2.2 + Web Tools Project (WTP 1.5.3 only)
- Toolkit also supports application development for the Lotus Expeditor runtime on these device operating systems:
  - ▶ Windows Mobile® 2003 SE and Windows Mobile 5 (PPC and Phone Editions)
  - ▶ Microsoft Windows CE 5 Professional and Microsoft Windows CE 4.2 (Micro Broker only)
  - ▶ Nokia E90

The Lotus Expeditor Toolkit runs on these platforms and supports application development for the Lotus Expeditor runtime on these same platforms:

- Microsoft Windows XP Service Pack 2
- Microsoft Windows Vista
- RedHat Enterprise Linux 4.0 WS with GTK support – Update 3 and 4
- SUSE Linux Enterprise Desktop, version 10 (SLED 10)
- RedHat Enterprise Linux 5.0 WS with GTK Support – Eclipse 3.2.2 + Web Tools Project (WTP 1.5.3 only)

The toolkit also supports application development for the Lotus Expeditor runtime on these device platforms:

- Windows Mobile 2003 SE and Windows Mobile 5 (PPC and Phone Editions)
- Microsoft Windows CE 5 Professional and Microsoft Windows CE 4.2 (Micro Broker only)
- Nokia E90

## One of these tools must be installed

- Rational Application Developer 7.0.0.2
- Rational Software Architect 7.0.0.2
- Eclipse 3.2.2 + Web Tools Project (WTP) 1.5.3
- WebSphere Application Server toolkit 6.1.1.2

In addition, Lotus Expeditor Toolkit can be installed into any of these development tools:

- Rational Application Developer 7.0.0.2
- Rational Software Architect 7.0.0.2
- Eclipse 3.2.2 + Web Tools Project (WTP) 1.5.3
- WebSphere Application Server toolkit 6.1.1.2

## Toolkit functions across tools platforms

Toolkit function	Rational Application Developer and Rational Software Architect 7.0	WebSphere Application Server toolkit 6.1.1	WTP 1.5.3
Client services	√	√	√
Platform builder	√	√	√
Web application	√	√	√
JSF	√	Subset of Rational development tools	
Portlet	√	√	
Web services	√	√	√
WS-Security editors	√	√	
Embedded transaction	√	√	
Samples gallery	√	√	

**Limitations when using Lotus Expeditor Toolkit on Eclipse + WTP:** This list describes the functionality of Lotus Expeditor Toolkit supported when Lotus Expeditor Toolkit is installed into Rational Application Developer, WebSphere Application Server toolkit, and Rational Software Architect, but not supported when Lotus Expeditor Toolkit is installed into Eclipse + WTP:

- Embedded Transaction Projects** - Eclipse + WTP does not allow for usage of the Embedded Transaction Container Tools. If you plan to use these tools, you must install Lotus Expeditor Toolkit into Rational Application Developer, WebSphere Application Server toolkit, or Rational Software Architect.
- Samples gallery** - The samples gallery is not part of Eclipse or WTP. Therefore, the provided samples are not accessible from a samples gallery.
- Portlet tools** - The portlet tools extend the portlet tools that are part of WebSphere Application Server toolkit and Rational Application Developer. Therefore, to use the portlet tools you must install Lotus Expeditor Toolkit into Rational Application Developer, WebSphere Application Server toolkit, or Rational Software Architect.
- WS-Security editors** - The WS-Security editors extend the Rational Application Developer security editors. You can build security enabled Web services for JSR-172 by hand, without the aid of the editors. To use the editors, you must install Lotus Expeditor Toolkit into Rational Application Developer, WebSphere Application Server toolkit, or Rational Software Architect.
- Full JSF** - Rational Application Developer and WebSphere Application Server toolkit do not include the WTP implementation of the JSF tools. Rational Application Developer will continue to provide a more advanced level of JSF support. Therefore, the usage of JSF with Lotus Expeditor will differ based on the level of base ware.

## Hardware requirements

- Minimum hardware requirements:
  - ▶ 512 MB RAM
  - ▶ 500 MB free disk space
  - ▶ x86 processors capable of supporting the supported operating systems
- Display resolution:
  - ▶ SVGA (1024 x 768) display minimum
- High contrast mode:
  - ▶ 1024 x 768 display minimum
  - ▶ Only default font settings (10pt Sans)

You must ensure that each machine on which you plan to install the IBM Lotus Expeditor 6.1.1 Toolkit meets the requirements shown on this slide.

## Agenda

- Introduction
- System requirements for the toolkit
- **Installing and configuring the toolkit**
- Using the Lotus Expeditor Toolkit with Lotus Expeditor
- Using the Lotus Expeditor Toolkit with Lotus Sametime
- Using the Lotus Expeditor Toolkit with Lotus Notes 8
- Using the toolkit for packaging and deployment
- References

## Installing the toolkit

Check to develop desktop and device applications on Lotus Expeditor

Check to develop desktop applications on Lotus Expeditor

Check these 3 features individually only in special situations

Check to develop device applications on Lotus Expeditor

On Toolkit CD, the toolkit update site is located at <CD directory>\Expeditor\_Toolkit\_install

15 Toolkit overview © 2007 IBM Corporation

In the 6.1.1 release, the components of the desktop toolkit have been separated into three features. The first of which is the Lotus Expeditor Tools. This feature should be installed by all toolkit users. The other two features available in the desktop toolkit are the Lotus Expeditor Development Runtimes VM (jclDesktop) and the Lotus Expeditor Development Runtimes. These features can be installed at the same time as the Lotus Expeditor Tools feature or later on as a separate update.

Users who are developing applications for the Lotus Expeditor for Desktop offering should install the Lotus Expeditor Development Runtimes feature in addition to the Lotus Expeditor Tools feature. Users who are developing applications for the Lotus Notes 8 or Lotus Sametime environments would not need to install the Lotus Expeditor Development Runtimes feature. Similarly, the Lotus Expeditor Development Runtimes VM (jclDesktop) can be installed by users who select this JRE for use in developing their applications. Users who want to use the Java 5 VM from the Device Runtimes Environment installation CD, would not need to install jclDesktop.

To install the toolkit, select **Help > Software Updates > Find and Install** in the IDE. For a first time install, you will select **Search** for new features to install and then press **Next**. You can then 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. Expand **desktop** and select the desktop toolkit features to 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 appears that shows you a summary of the features that will be installed. Press **Finish** to perform the installation. You will receive a prompt as each of the features 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.

IBM Software Group | Lotus Expeditor 6.1.1 Education IBM

## Configuring the toolkit

- Configuration dialog allows you to select:
  - ▶ Lotus Expeditor for Desktop
  - ▶ Lotus Expeditor for Device
  - ▶ Lotus Notes 8
  - ▶ Lotus Sametime
- Configures
  - ▶ Target platform
  - ▶ Default workspace VM
  - ▶ Compiler compliance level
- Control when prompting occurs
  - ▶ Every time
  - ▶ Once per workspace
  - ▶ Never

**Lotus Expeditor Toolkit Configuration**

Test Environment Configuration

Test Environment: Lotus Expeditor for Desktop

Target Location:  Browse

VM Name: jcdesktop Win32 x86

Compiler compliance level: 1.4

Auto-Configuration Preference

Attempt to configure the toolkit each time a workspace opens

Attempt to configure the toolkit the first time a workspace opens

Do not attempt to auto-configure the toolkit

Restore Defaults

OK Cancel

Changing Target Platform to /C:/eclipse3.2/eclipse/plugins/com.ibm...

Operation in progress...

org.w3c.sac\_1.0.0.0-20060714

Run in Background Cancel Details >>

16 Toolkit overview © 2007 IBM Corporation

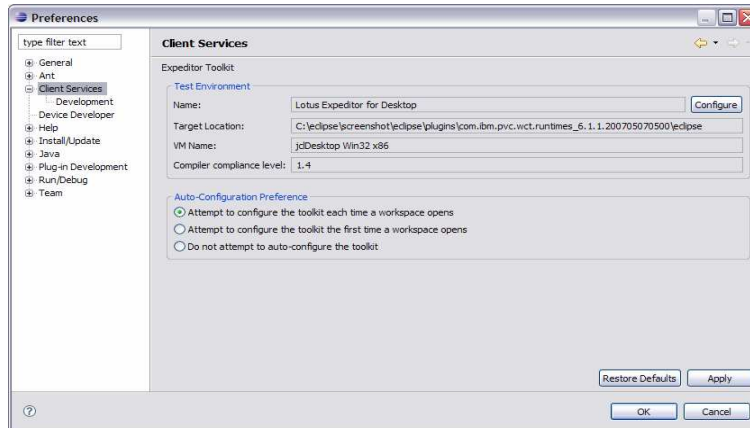
When the IDE starts up for the first time after the toolkit is installed, you will need to set preferences for the toolkit before you start developing applications. Configuration assistance is given in the form of a pop-up dialog on startup. The pop-up will provide you with a list of known test environments. Selecting one of these test environments populates the remaining fields in the dialog (target location, VM name and compiler compliance level) with default values that are appropriate for use with the particular test environment. When **OK** is pressed in this dialog, the required toolkit configuration steps will be performed for you.

In addition to setting the Eclipse target platform, the default JRE and compiler compliance level, you can also control when the configuration dialog appears. It can be displayed each time a workspace starts, the first time a workspace starts, or never.



## Re-configuring the toolkit

- Use **Window > Preferences > Client Services** option to modify the configuration



You can change the configuration settings for the toolkit by invoking the configuration dialog from the **Window > Preferences > Client Services** preference page. You might do this, for example, if you are developing for the desktop client and want to also develop a device client application.

## Agenda

- Introduction
- System requirements for the toolkit
- Installing and configuring the toolkit
- **Using the Lotus Expeditor Toolkit with Lotus Expeditor**
- Using the Lotus Expeditor Toolkit with Lotus Sametime
- Using the Lotus Expeditor Toolkit with Lotus Notes 8
- Using the toolkit for packaging and deployment
- References

## Development steps

- Create a client services project
- Select a target definition for the project
- Edit and compile the project
- Create a client services launch configuration
- Debug and test the project by using the launch configuration
- Assemble project for deployment

## Client services tools

- Extends Eclipse Plug-in Development Environment (PDE) tools
- Provides project creation and configuration
- Application templates
  - ▶ Basic client services
  - ▶ Rich basic application
  - ▶ Preference page basic application
- Automated manifest management
- Provides a set of PDE target definitions
- PDE style launch
  - ▶ Java SE, jclDesktop and jclDevice configurations
  - ▶ Launch using selected target definition

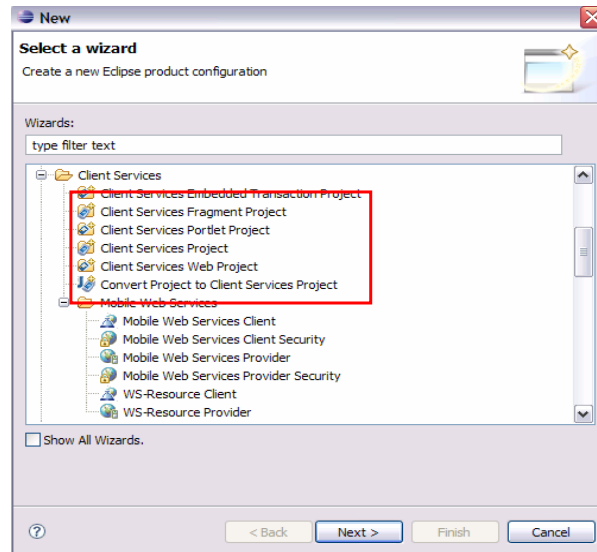
The Bundle Developer Kit extends Eclipse's PDE tools and is the component in the toolkit responsible for the creation of Client Services projects.

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.

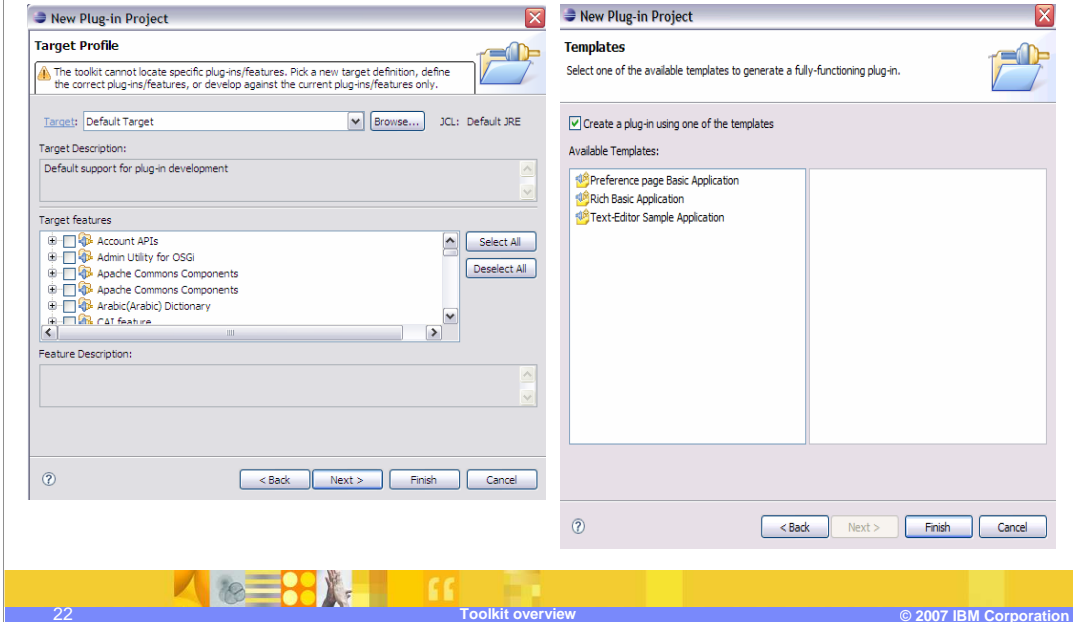
Finally, the Bundle Developer Kit provides a PDE style launch that can be used to test applications. PDEs launch configuration has been extended to enable selection of the Target Definition, JRE and personality.

## Client services project wizards



On this slide you can see a list of the types of Client Services project that are supported in the toolkit. There are Client Services projects and fragment projects, Embedded Transaction projects, Web Projects, Web Services Projects and Portlet Projects.

## Project creation – Client services project



When you are creating a Client Services project, one of the wizard pages that you will see allows you to select the Target definition and 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. Simply click **Finish** in this dialog.

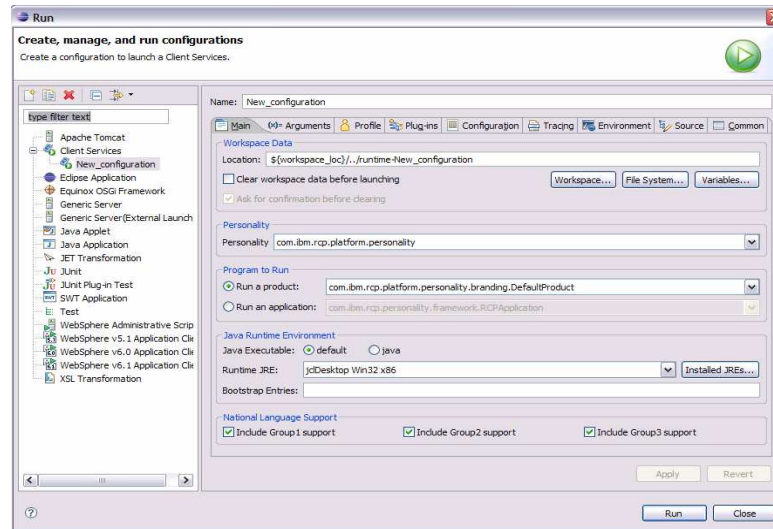
## Comparing client services and plug-in projects

- Both client services projects and plug-in projects can be used to produce components that will run on the Lotus Expeditor runtime
- Client Services projects are plug-in projects, so you can use the full PDE tools on client services projects
- Client services projects add these additional tools:
  - ▶ Automatic maintenance of the manifest to reflect project dependencies
  - ▶ Use of target definitions to allow selection of project dependencies at a descriptive feature level, rather than the plug-in level
  - ▶ Use of target definitions to properly set the project's Java build path and JRE to match the targeted Expeditor platform

**Client Services Project:** A Client Services project contains a set of plug-ins and an associated Target Definition. All or some of the Target Definition's features may be selected. In addition, Client Services projects can:

- Automatically update the Java Build Path. Lotus Expeditor Toolkit can automatically update the project's Java Build Path to reflect the project's Target Definition and Features settings.
- Provide a default plug-in activator. The toolkit can create a default bundle activator class. You can tailor the default plug-in activator class by editing the source file for the class folder.
- Automatically update the Manifest file. The toolkit can automatically update the Manifest file in the Client Services project to contain appropriate OSGi metadata for the project. Client Services manages or provides initial default values for the metadata fields by:
  - Setting Bundle-Name to the project name on project creation – Setting Bundle-Version to 1.0.0 on project creation
  - Setting Bundle-Activator to the default bundle activator if one was created
  - Updating Import-Package and Require-Bundle to reflect the packages imported by the project's classes.

## Launching



To configure the Client Services launcher:

1. From the Java perspective, select **Run > Run...** to invoke the Launch Configuration Dialog wizard
2. Create a new Client Services launch configuration. Right click **Client Services**. From the pop-up menu, select **New**. At this point, you can run this configuration with default values or proceed to configure it further.
3. In the **Main** tab of this new Client Services launch configuration, select the **Workspace Data**, **Personality**, **Program to Run**, **Java Runtime Environment** and **National Language Support** groups for this launch.
4. The **Argument** tab contains default values for the Program and VM argument data of this Target Definition.
5. From the **Target** tab, select any necessary pre-defined Target Definitions for this launch.
6. In the **Plug-ins** tab, individual plug-ins may be selected in addition to the list already defined in the **Target** tab.
7. The **Configuration** tab contains default configuration data for the location and format of this configuration file.
8. The **Tracing** tab displays a list of plug-ins that support tracing.
9. Use the **Environment** tab to define environment variables for the configuration.
10. Use the **Source** tab to add any additional sources or archives to the configuration.
11. The **Common** tab contains default configuration data for the OSGi console.
12. Click **Run** to launch the new Client Services launch configuration.



## Target definitions

- Replacement for platform profiles
- Provide grouping of features or plug-ins in the client platform
  - ▶ Default Target (desktop)
  - ▶ Default device Target (device and desktop)
  - ▶ Default with embedded Sametime target (desktop)
  - ▶ Notes 8 target (desktop)
  - ▶ Sametime target (desktop)
- Also defines related product, application, VM and personality
- Users can add target definitions

**Target Definition:** A target definition specifies all aspects of a target - including its location, constituent plug-ins and environment. Refer to the Eclipse Documentation for more information on Target Definitions. The Client Services project wizard provides a method for defining the Target's features and each feature's plug-ins to be used at runtime. Note that each Target's feature is composed with a set of one or more plug-ins.

A Target Definition defines a set of Target features. Target features enable you to focus on the logical service requirements of the service instead of the requirements of the actual underlying Client Services plug-ins.

When you create a Client Services project, you select a Target Definition and a set of Target features for the project. The Lotus Expeditor Toolkit updates the Java Build Path for the project to reflect the Java Runtime Environment (JRE) of the platform, based on the Target Definition and Target features you selected. The Target Features you selected are automatically added to the Java Build Path of the project.

The Lotus Expeditor Toolkit provides Target Definitions that represent the bundles available in Lotus Expeditor, Notes® 8 and Sametime 7.5.1 runtimes.

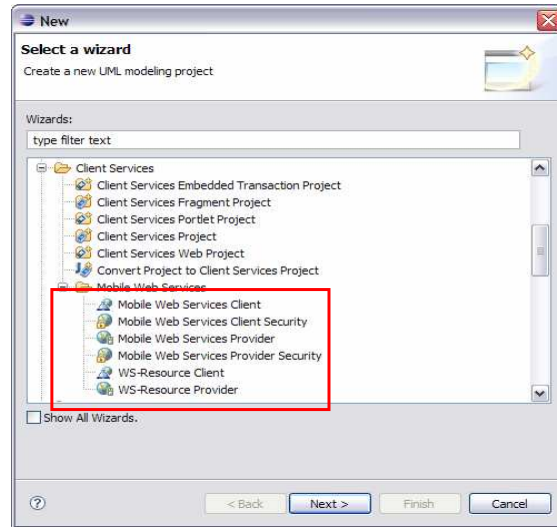
The Lotus Expeditor Toolkit has extended the functionality of a regular PDE target definition. The additional attributes and elements created by the Lotus Expeditor Toolkit are used to modify the behavior of the Client Services Launcher. A target can contain a list of feature and plug-in elements. These elements are displayed on the **Target** tab of the Client Services Launcher.

A target definition may also specify the config.ini location to be used during launch. You can do this by adding a configuration element with a location attribute. The location can be absolute or contain variables. The value of the location will set the location of the config.ini template to be used for launching. A default personality can also be specified with a personality element with an ID attribute. The value of the ID attribute should be the personality ID to be specified during launch.

Additional launcher settings include specifying environment variables.

Finally, a target definition can define the default product and application for launching. This is specified in a program element. You can replace the default Lotus Expeditor branding with a custom branding, by replacing the Default Branding feature (com.ibm.rcp.personality.default.branding.feature) with your own feature. Branding is the appearance and user experience of the product - it includes splash screens and window decorations.

## Mobile Web services wizards



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

## Mobile 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 both generate client code that consumes Web Services and to expose OSGi services as Web Services providers. An application that consumes 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 edit your Web Services security configuration for the client and provider. These editors are available in Rational Application Developer, Rational Software Architect and the WebSphere Application Server toolkit environments.

In addition to developing Mobile Web Services, the toolkit can 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 application tools

- Construct Web applications
  - ▶ JSP, Servlet, JSF, JSTL and Struts support
  - ▶ Deploy and run on the client runtime
- Convert or re-target
  - ▶ Run and debug Expeditor applications on WebSphere Application Server
  - ▶ Run and debug WebSphere applications on Expeditor

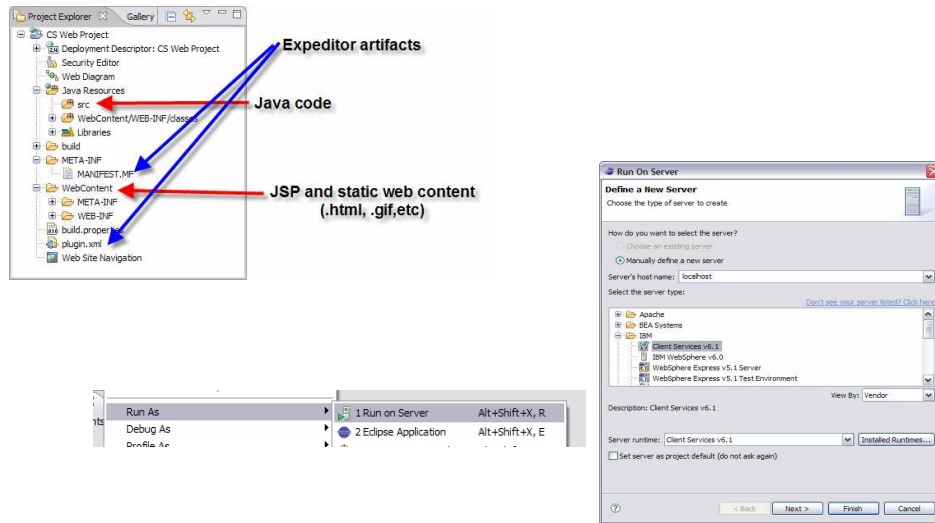
Web Application tools enable development of Client Services Web applications that use JSPs, servlets, Struts, Java Server Faces (JSF) and JSTL which are JSP standard tag libraries. When creating an application for a device, Struts, JSF and JSTL are not supported.

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 will be 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



On this slide, in the upper left, you see 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. On the bottom 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, select Client Services V6.1 as the server type.

## Client services Web projects versus dynamic Web projects

- Client services Web projects ARE dynamic Web projects, with extensions to aid development for the Expeditor runtime
- In particular, client services Web projects are both dynamic Web projects and plug-in projects
  - ▶ You can use the full set of dynamic Web development tools and plug-in development tools on the project
  - ▶ You can run and test the project on an Expeditor runtime and on a different runtime, such as WebSphere Application Server
  - ▶ You can export the project as a WAR file (for a standard J2EE server such as WebSphere Application Server) or export the project as a deployable plug-in, feature, or update site for Expeditor

## Portlet tools

- Construct JSR168 compliant portlet applications
  - ▶ Faces, Basic and Empty portlet type support
  - ▶ Deploy and Execute on client runtime
- Convert or re-target
  - ▶ Run & Debug WebSphere Portal applications on Expeditor
  - ▶ Run & Debug Expeditor applications on WebSphere Portal
- Limited Portal type functional support
  - ▶ Aggregation of portlets in JSP is supported
  - ▶ Portlet communication using WSDL based wiring is supported

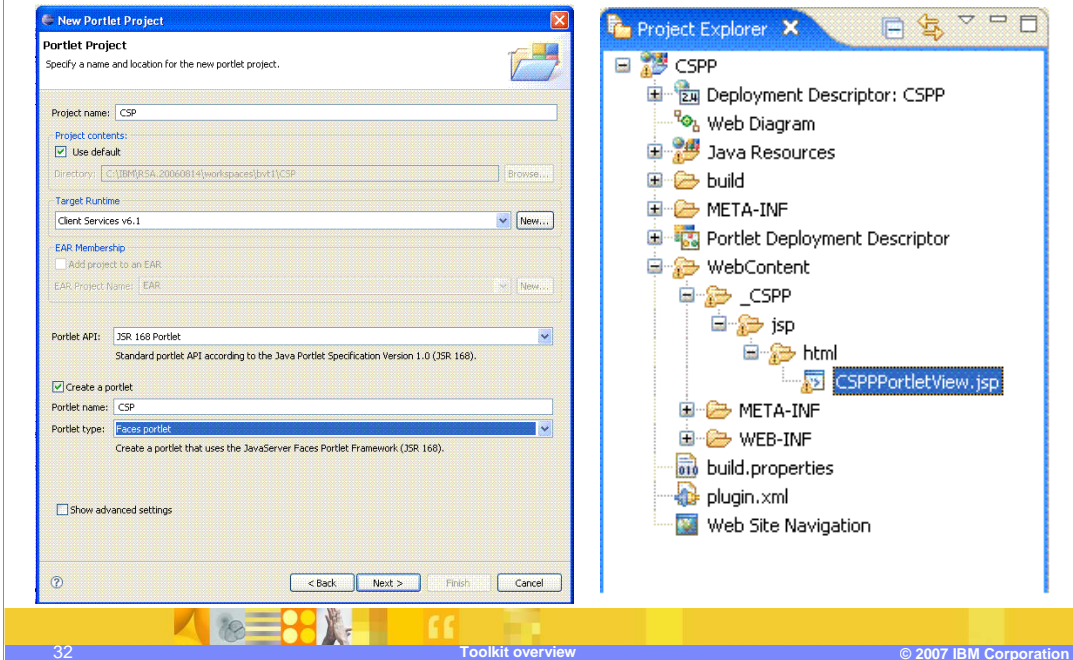
The Portlet tools were new in the 6.1 release. These tools allow JSR 168 compliant portlet applications to be constructed. When creating a Client Services portlet project, you can select the portlet type from a drop down menu in the wizard. Empty, Basic and Faces portlet types are supported. The portlet 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 Portal Server based JSR 168 portlet application into a Client Services portlet project. And, you can also run and debug WebSphere Portal 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.

The Portlet aggregation function is not available on WebSphere Application Server toolkit.

## Portlet project



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



## Embedded transaction tools

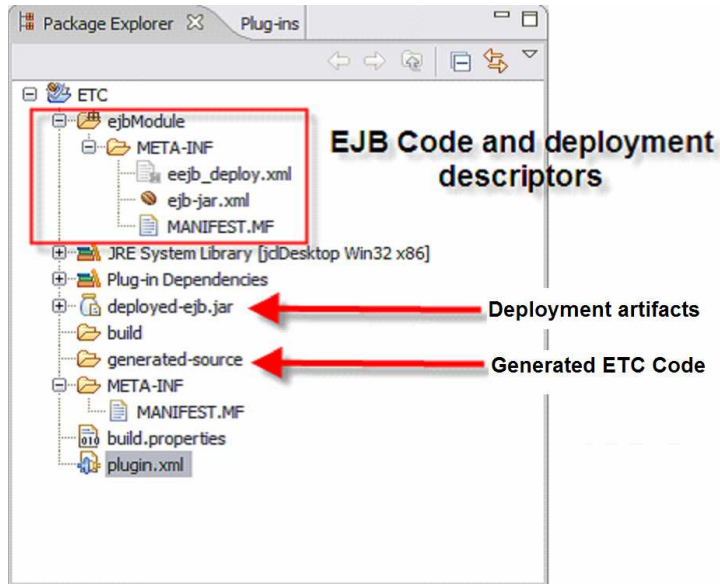
- ▶ The runtime supports a subset of EJBs
  - Stateless Session Beans
  - Entity Beans (BMP and CMP)
- ▶ Construct runtime compliant EJBs
  - Creating new client services projects
  - Converting existing Java EE EJB projects
- ▶ Usage of functionality not supported by runtime reported
  - During conversion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

The Embedded Transaction Tools are designed to support the subset of EJBs that are supported in the runtime. There is support for both stateless session beans and 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 and the 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, project validators detect functionality that is not supported by the runtime and reports these as errors.

## Embedded transaction project



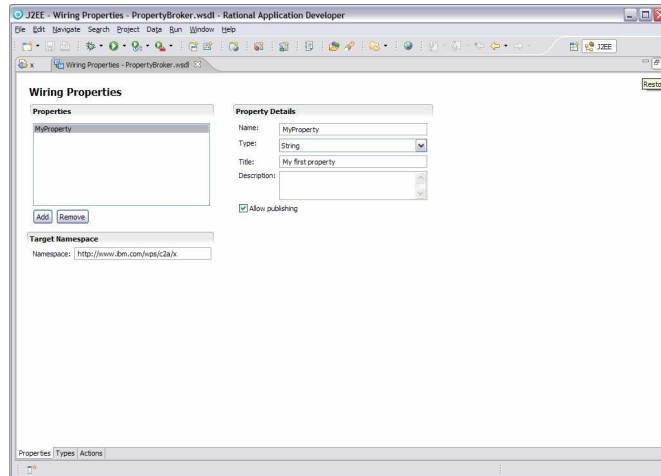
This slide gives an example of the contents of an Embedded Transaction project. The same Lotus Expeditor artifacts are here as with the other Client Services project types, and you can see the location where the source and generated code reside in the project. The deployed-ejb jar file contains EJB binaries and properties required by the ETC runtime.

## Client services portal, embedded transaction and mobile Web services

- Client services portal projects are similar to client services Web projects in that you can use them to develop for the Lotus Expeditor runtime and other runtimes
- Since embedded transaction projects are a subset of EJBs, they typically require modification in order to target runtimes other than Lotus Expeditor
- Mobile Web services generate stub code that is specific to the Expeditor runtime, and cannot be deployed to non-Expeditor runtimes without re-generating the stub code

## Property broker editor

- Define wires for property broker communication
  - ▶ Portlet to Portlet
  - ▶ Portlet to Eclipse
  - ▶ Eclipse to Portlet
- 3 Tab editor
  - ▶ Define properties
  - ▶ Define types
  - ▶ Define actions



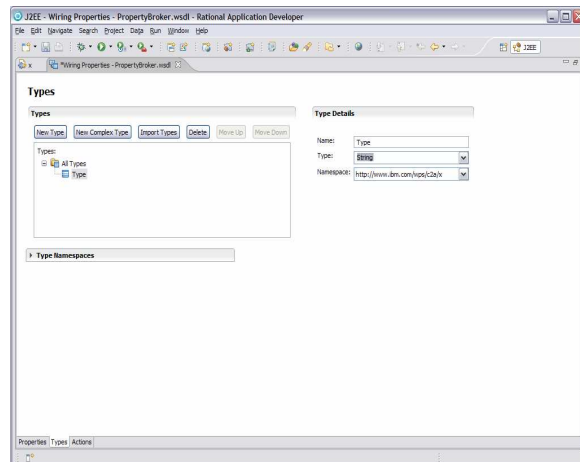
WSDL files that are used in composite applications are created and edited using the Property Broker Editor feature. This feature provides an easy way for you to describe property broker properties, actions, and types without having to understand the WSDL format. You define the properties, actions, and types through a simple interface which then generates the WSDL file required by property broker implementations when you save the editor settings. The editor can be launched using either of these methods:

-Create a new Property Broker WSDL file using its New File wizard, by selecting **File > New > Client Services > Wiring Properties**.

-Edit an existing file, by selecting it in the Navigator, then launching the editor using **Open With > Wiring Properties**.

## Property broker editor - Types

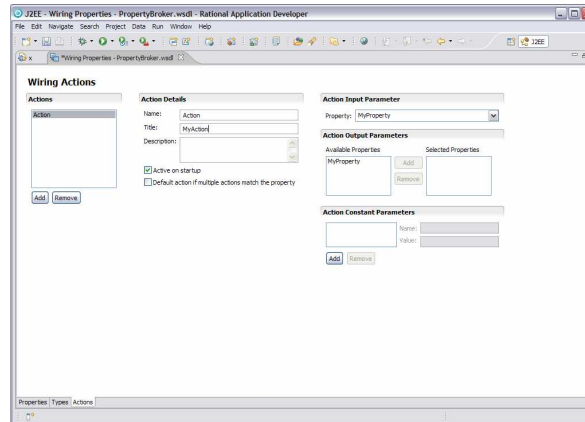
- With composite applications, a data type is associated with the properties that are used to exchange information between application components
- The property broker transfers instances of these properties and their values to and from components to initiate actions
- New data types can be created to supplement the defined ones:
  - ▶ String
  - ▶ Integer
  - ▶ Double
  - ▶ Boolean
  - ▶ Date
  - ▶ Time
  - ▶ Date/Time



With composite applications, properties are used to exchange information between application components using the property broker. The property broker transfers instances of these properties and their values to and from portlets to initiate actions. The property broker is a WebSphere Portal interface that negotiates the relationship between cooperative source and target portlets and delivers data to the target in the form of the corresponding portlet action. Use these steps to set data types for these properties with the Property Broker Editor feature:

## Property broker editor - Actions

- With composite applications, you need to define actions that are triggered when portlets publish properties that get received by another portlet
- Examples of actions include 'changing' or 'refreshing' component pages and redirecting current page to a Web site
- These steps must take place before an action can occur:
  1. User interacts with a portlet for example, inputs a value and clicks a submit button
  2. A script runs and publishes property value to property broker
  3. Property broker finds portlets that are listening for a property value and tells them to perform the requested action



With composite applications, you need to define the actions that are triggered when portlets publish properties that get received by another portlet. Examples of actions include changing or refreshing component pages and redirecting the current page to a Web site. In composite applications, these steps must take place before an action can occur.

1. The user interacts with a portlet in some way, for example, inputting a value and then clicking a submit button.
2. A script runs and publishes the property value to the IBM WebSphere Portal property broker.
3. The property broker finds the portlets that are listening for the property value and tells them to perform the requested action.

## Debugging support

- Client services projects leverage the same Eclipse symbolic debugging support available to Java projects
- JSP source can be debugged symbolically, including breakpoints, and variable inspection
- The components can be tested in the Lotus Expeditor runtime through the tool's client services launcher, in both run and debug mode
- J2EE components can also be tested through the *Run on Server* and *Debug on Server* actions
- An external runtime can be debugged provided it was launched with the proper debug options

## 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, 6.0 and 6.1 projects are migrated for most components
- There are limitations such as:
  - ▶ Web applications and Embedded Transaction applications created using Rational Application Developer or Rational Software Architect are not migrated in an Eclipse 3.2.2 or WebSphere Application Server toolkit 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 V6.1.1 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 6.1.1. 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 V6.1.1 tools components migrate projects from the 5.7.1, 6.0 and 6.1 releases. Platform Builder migrates only projects from the 6.0 and 6.1 releases since it does not have support for devices. In addition, Web applications and Embedded Transaction projects created with Rational Application Developer or Rational Software Architect will not be migrated in an Eclipse 3.2.2 or WebSphere Application Server toolkit environment.



## Agenda

- Introduction
- System requirements for the toolkit
- Installing and configuring the toolkit
- Using the Lotus Expeditor Toolkit with Lotus Expeditor
- **Using the Lotus Expeditor Toolkit with Lotus Sametime**
- Using the Lotus Expeditor Toolkit with Lotus Notes 8
- Using the toolkit for packaging and deployment
- References

## Configuring for Lotus Sametime

Two methods for developing Lotus Sametime applications

- Target a Lotus Sametime Connect client
  - Select Lotus Sametime in the toolkit configuration dialog box
- Target an embedded version of Sametime Connect client running within the Lotus Expeditor desktop
  - Select Lotus Expeditor for Desktop in the toolkit configuration dialog box
  - Change Default Target Selection for workspace to Default with Embedded Sametime Target
    - Window > Preferences... > Client Services > Development > Default Target Selection



There are two methods for developing Lotus Sametime applications provided in the Lotus Expeditor Toolkit.

First, you may target a Lotus Sametime Connect client installation on your development machine by selecting the Lotus Sametime Test Environment when opening a new workspace. Second, you can select the Lotus Expeditor for Desktop Test Environment when opening a new workspace, and then specifically select the Default with Embedded Sametime Target for your projects and Run configurations.

**Configuring for Lotus Sametime** After installing the Lotus Expeditor Toolkit and restarting the IDE, you are presented with a dialog box for toolkit configuration. From the Test Environment drop down box, select **Lotus Sametime**. The rest of the fields are updated based on the default information for Lotus Sametime. If you have not installed Lotus Sametime into the default location, you can browse to the appropriate Target Location for your system before pressing **OK**. Note that in addition to setting the target platform, JRE and compiler compliance level, the toolkit's configuration process also sets the default target to Lotus Sametime. This causes the target field to be set to Lotus Sametime in all dialogs where the target name must be given.

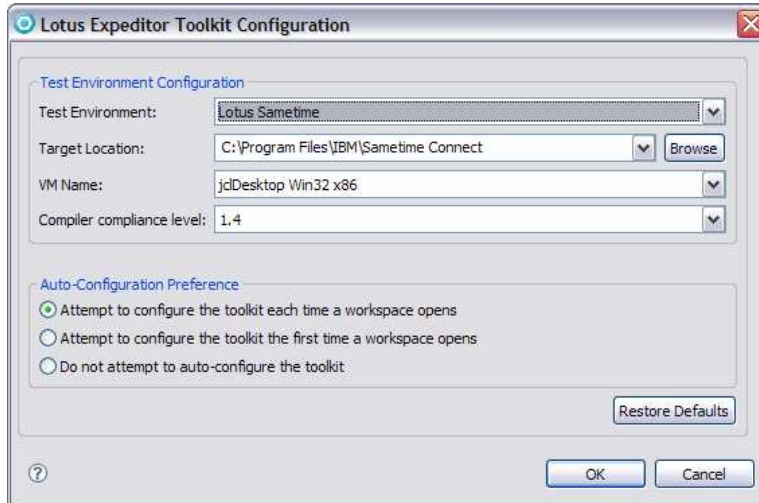
The recommended setting for the auto-configuration preference is **Attempt to configure the toolkit the first time a workspace opens**. If you select this radio button, you will only be prompted to configure the toolkit the first time you use a workspace with the toolkit. When the toolkit is configured, the target platform, JRE, compiler compliance level and default target definition are selected.

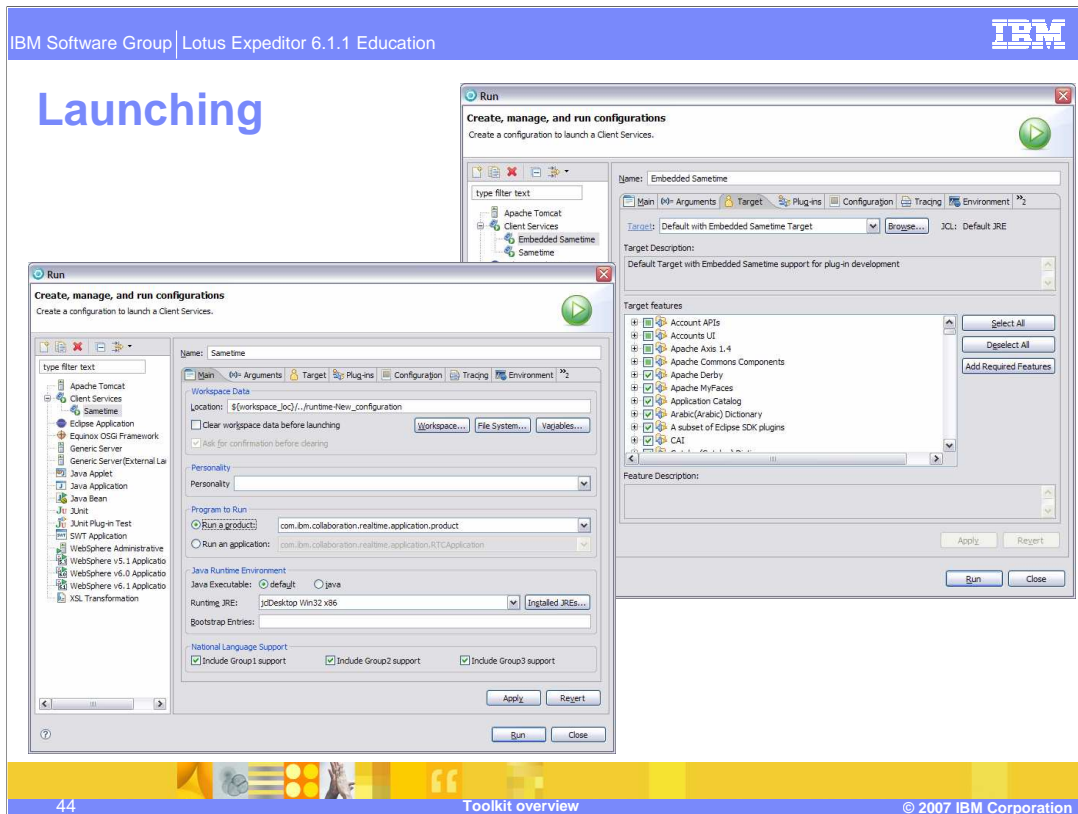
**Configuring for Lotus Expeditor with Embedded Sametime** After installing the Lotus Expeditor Toolkit and restarting the IDE, you are presented with a dialog box for toolkit configuration. From the Test Environment drop down box, select **Lotus Expeditor for Desktop**. The rest of the fields are updated based on the default information for Lotus Expeditor. Select **OK** to close the dialog window and complete the toolkit configuration. The recommended setting for the auto-configuration preference is **Attempt to configure the toolkit the first time a workspace opens**. If you select this radio button, you will only be prompted to configure the toolkit the first time you use a workspace with the toolkit. When the toolkit is configured, the target platform, JRE, compiler compliance level and default target definition are selected.

After selecting the Lotus Expeditor for Desktop Test Environment, you must also change the Default Target Selection for your workspace to **Default with Embedded Sametime Target**:

1. From within the Eclipse IDE, select **Window > Preferences**.
2. Expand **Client Services**.
3. Select **Development**.
4. From the Default Target Selection control, select the Default with Embedded Sametime Target.
5. Click **OK**.

## Toolkit configuration dialog box





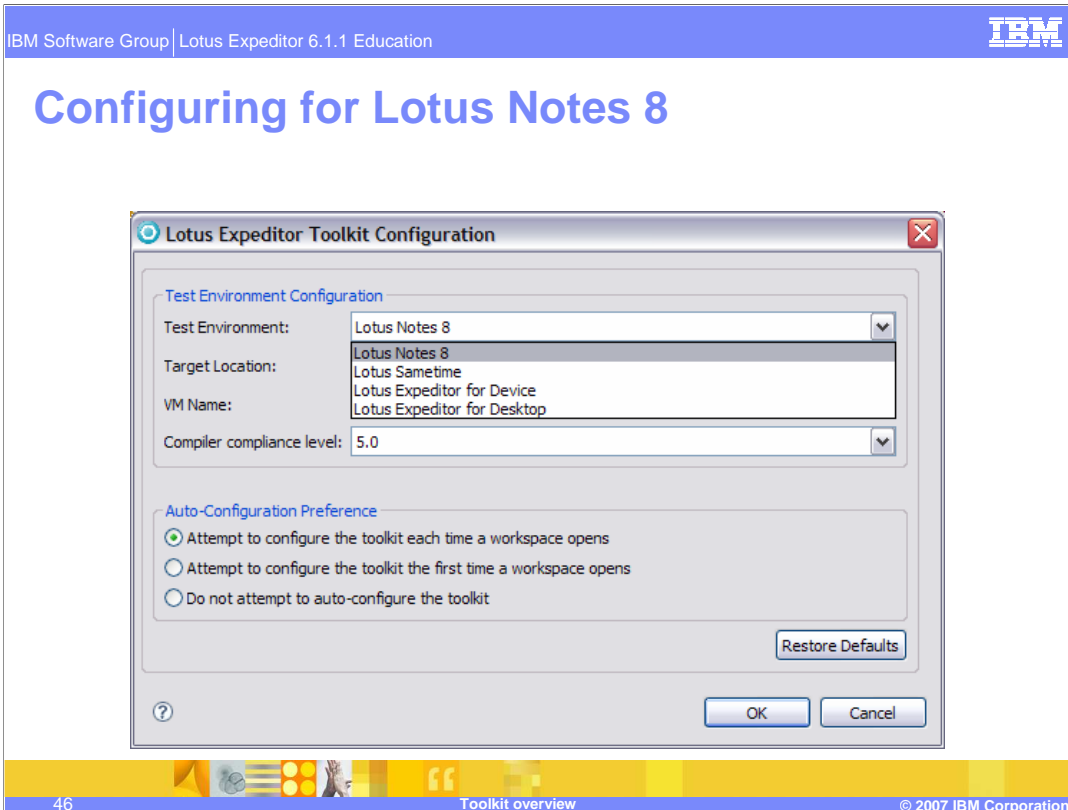
After developing your applications, you can launch Lotus Sametime or Lotus Expeditor with Embedded Sametime to test them from within the IDE. Perform these steps to create a launch configuration and launch Lotus Sametime or the Lotus Expeditor with Embedded Sametime platform from within the IDE:

1. Select **Run > Run...**
2. Select **Client Services**.
3. Click **New**.
4. Enter a Name.
5. Select **Clear workspace data before launching**.
6. Click **Run**.

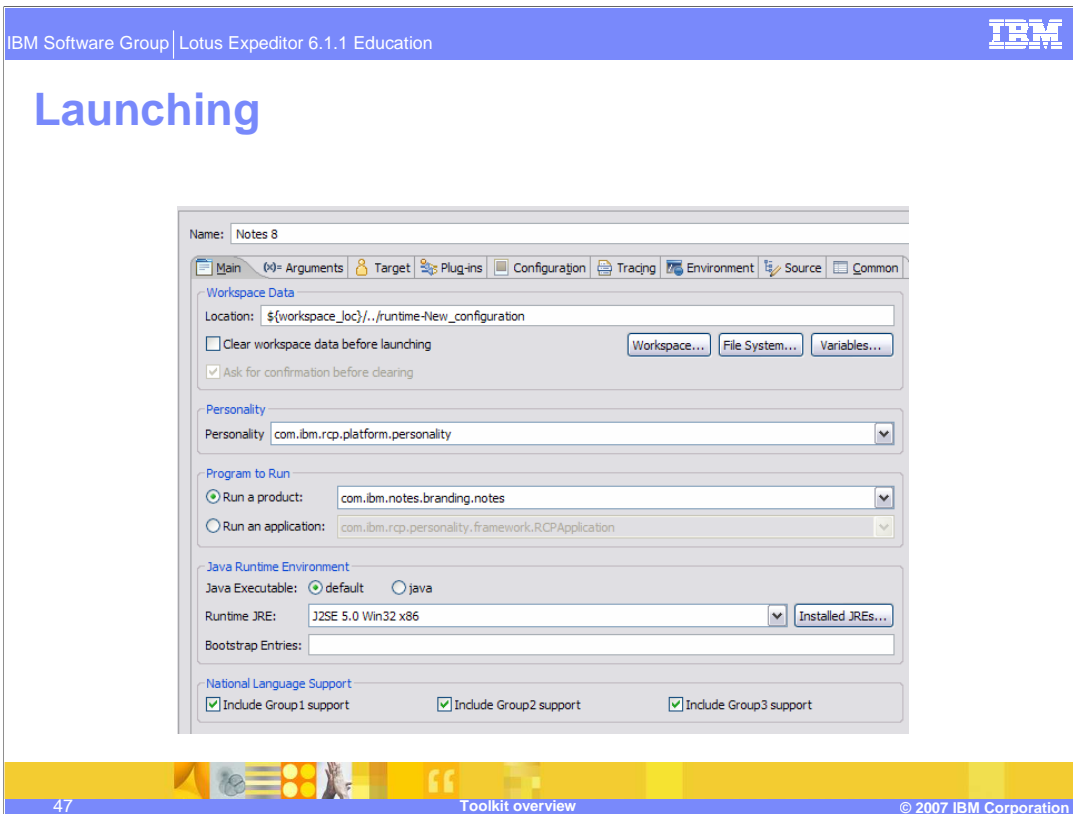
The Lotus Sametime and Lotus Expeditor for Desktop target set the VM arguments for the launch configuration. The tabs on the client services launcher function similar to the Eclipse PDE launcher, with a few additions. If you want to add or remove plug-ins from the launch configuration at a feature level, change to the target tab and select the features to add or remove.

## Agenda

- Introduction
- System requirements for the toolkit
- Installing and configuring the toolkit
- Using the Lotus Expeditor Toolkit with Lotus Expeditor
- Using the Lotus Expeditor Toolkit with Lotus Sametime
- **Using the Lotus Expeditor Toolkit with Lotus Notes 8**
- Using the toolkit for packaging and deployment
- References



After installing the Lotus Expeditor Toolkit and restarting the IDE, you will be presented with a dialog box for toolkit configuration. From the Test Environment drop down box, select **Lotus Notes 8**. The rest of the fields will be updated based on the default information for Lotus Notes 8. If you have not installed Lotus Notes 8 into the default location, you can browse to the appropriate Target Location for your system before pressing **OK**. Note that in addition to setting the target platform, JRE and compiler compliance level, the toolkit's configuration process also sets the default target to **Notes 8 Target**. This causes the target field to be set to **Notes 8 Target** in all dialogs where the target name must be given. The recommended setting for the auto-configuration preference is **Attempt to configure the toolkit the first time a workspace opens**. If you select this radio button, you will only be prompted to configure the toolkit the first time you use a workspace with the toolkit. When the toolkit is configured, the target platform, JRE, compiler compliance level and default target definition will all be selected.



After developing your applications, you can launch Lotus Notes 8 to test them from within the IDE. To create a launch configuration and launch Lotus Notes 8 from within the IDE, perform this procedure:

1. Select **Run > Run...**
2. Select **Client Services**.
3. Click **New**.
4. Enter a Name.
5. Select **Clear workspace data before launching**.
6. Click **Run**.

The Notes 8 Target sets the VM arguments for the launch configuration. The tabs on the client services launcher function similar to the Eclipse PDE launcher, with a few additions. If you want to add or remove plug-ins from the launch configuration at a feature level, change to the profile tab and select the features to add or remove.

## Agenda

- Introduction
- System requirements for the toolkit
- Installing and configuring the toolkit
- Using the Lotus Expeditor Toolkit with Lotus Expeditor
- Using the Lotus Expeditor Toolkit with Lotus Sametime
- Using the Lotus Expeditor Toolkit with Lotus Notes 8
- Using the toolkit for packaging and deployment
- References



## 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 the operating system of the client platform
- Starts with a list of features or plug-ins in target definition
- User can add from the target platform or workspace
- Supports inclusion of language groups and VM
- Allows test and export of configuration before deployment

The Platform Builder component allows you to package your applications along with features & 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.

As of the 6.1 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 VM, 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 6.1.1 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'.
- File name:** A text field contains 'pbjproduct'.
- Initialize the file content:**
  - Create a configuration file with basic settings
  - Use an existing product: com.ibm.esupport.client.Product
  - Use a launch configuration:
- 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: Overview | Configuration | Launcher | Branding

On this slide, there are screen captures of the product configuration wizard and editor. The wizard is shown on the left, and the editor on the right. The editor is a multi-page editor with an Overview, a Configuration, a Launcher and a Branding page.

## Packaging applications for distribution

- **Plug-ins (required)**
  - ▶ Plug-ins provide the core logic capability for the application, but they must be grouped into features in order to be installed with the Update Manager
  - ▶ Use Lotus Expeditor Toolkit to create plug-ins for Lotus Expeditor
- **Feature (required)**
  - ▶ Packages plug-ins into an installable unit
  - ▶ Use Plug-in Development Platform wizards to create Features
- **Update site (required)**
  - ▶ Packages features into an application
  - ▶ Use Plug-in Development Platform wizards to create Update Sites
- **Installer/Uninstaller (optional)**
  - ▶ A program to handle installation (and uninstallation of your application)
  - ▶ May be needed if you need to do anything more than provide an update site to allow installation of your application

**Plug-ins:** Plug-ins provide the core logic capability for the application, but they must be grouped into features in order to be installed with the Update Manager.

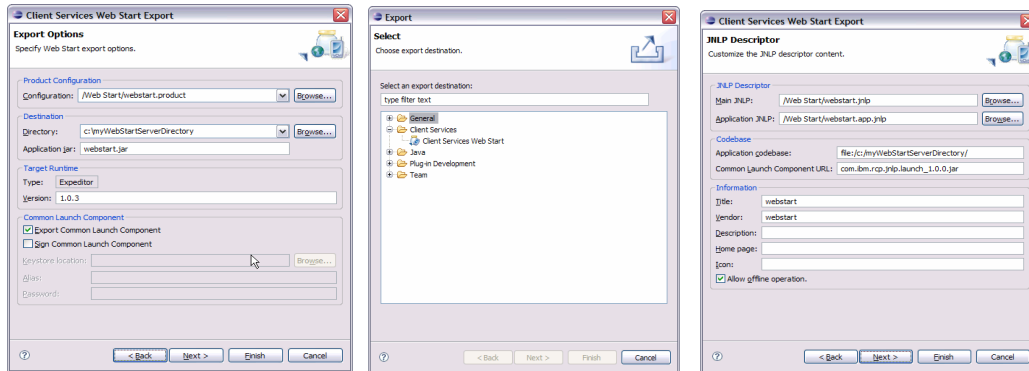
**Features:** A feature is the only level of installable unit that exists. You cannot choose to install only certain plug-ins from a feature. The Plug-in Development Platform provides wizards for creating Features.

**Update site:** An Update Site is the key mechanism to enable installation of the application.

**Installer/Uninstaller:** A program to handle installation (and uninstallation of your application) may be needed if you need to do anything more than provide an update site to allow installation of your application.

## Web Start

- Package an application for deployment using Web Start
  - Platform deployed to the client
  - Application deployed from server using Web Start



## Agenda

- Introduction
- System requirements for the toolkit
- Installing and configuring the toolkit
- Using the Lotus Expeditor Toolkit with Lotus Expeditor
- Using the Lotus Expeditor Toolkit with Lotus Sametime
- Using the Lotus Expeditor Toolkit with Lotus Notes 8
- Using the toolkit for packaging and deployment
- References

## References

- Developing applications for Lotus Expeditor at <CD directory>\doc\Application\_Developers\_Guide.pdf

## Feedback

### Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject= Feedback about xpdv6.1.1\\_toolkit\\_overview.ppt](mailto:iea@us.ibm.com?subject= Feedback about xpdv6.1.1_toolkit_overview.ppt)

You can help improve the quality of IBM Education Assistant content by providing feedback.

## 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 Lotus Notes Rational Sametime WebSphere

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

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

EJB, J2EE, Java, JRE, JSP, 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 NON-INFRINGEMENT. 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 2007. 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.

