



IBM Software Group

Add Your Own Tools to WDS*c* V5.0: Eclipse Plug-in Development

Don Yantzi (yantzi@ca.ibm.com)

IBM Toronto Lab

WebSphere software

@business on demand software

IBM eServer iSeries Technical Conference

Las Vegas, Nevada

June 23 – 27, 2003

IBM Software Group | WebSphere software



Disclaimer

- **Acknowledgement:**
 - ▶ This presentation is a collaborative effort of the IBM Toronto AS/400 Application Development presentation team, including work done by:
 - Phil Coulthard, George Farr, Claus Weiss, Don Yantzi, John Steinbacher, David Slater, Alison Butteril, Linda Cole
- **Disclaimer:**
 - ▶ The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.
- **Reproduction:**
 - ▶ The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.

Agenda

- Brief Introduction to Eclipse
- WebSphere Studio Workbench
- Extending the Workbench
 - ▶ Writing Eclipse plug-ins
 - ▶ HelloWorld plug-in example
- WebSphere Development Studio Client
 - ▶ iSeries Tooling in the Workbench
 - ▶ Extending WDS
- Summary



What is Eclipse?

WebSphere software



What is Eclipse?



- Eclipse is an empty slate:
 - ▶ Except for the Java tools it supplies
 - ▶ Use it to build AD products
 - By writing "plug-ins" that offer tools for particular programming needs
 - Plug-ins "snap into" Eclipse seamless
 - Eclipse + useful plug-ins = product
 - ▶ Common base for all new IBM AD products
 - ▶ Common base for many new 3rd party products

- Eclipse is an opportunity
 - ▶ To build and sell plug-ins for all eclipse users
 - All users of Eclipse and Eclipse-based products



Eclipse Open Source Organization

- ★ An open source community chartered to manage the evolution of an open application development integration platform
- ★ Managed by an independent, multi-vendor organization
- ★ Technology licensed via the Common Public License
- ★ Based on Java with initial support for Linux & Windows
- ★ Broad Industry Support



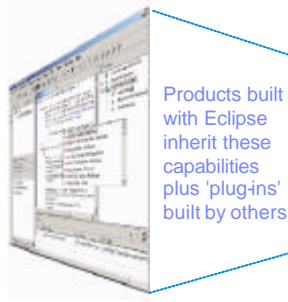
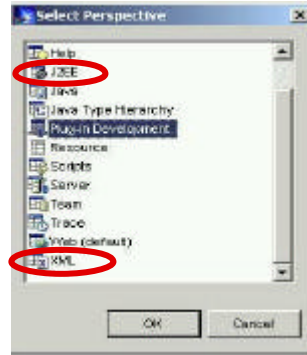
www.eclipse.org



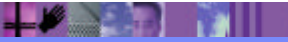
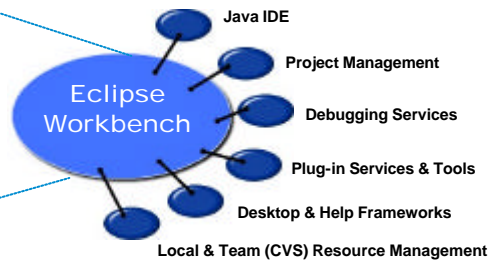
Eclipse Workbench

A 'PortalLike' Tool Integration Platform

A personalized, extensible development platform organized via developer "Perspectives"



Products built with Eclipse inherit these capabilities plus 'plug-ins' built by others



Eclipse - Broad Industry Support

Over 135 companies have been involved in the Eclipse project, including these industry leaders ...

- | | | |
|-------------------|--------------|-------------|
| Altoweb | Neuvis | Aldon |
| Bowstreet | QSSL | MKS |
| BrowserSoft | Rational | Softlanding |
| CommerceQuest | Red Hat | Lansa |
| Computer Assoc. | Serena | ASC |
| Compuware | Sitraka | |
| Crystal Decisions | Skyva | |
| Embarcadero | TeleLogic | |
| Holosofx | TogetherSoft | |
| Instantiations | Versant | |
| LegacyJ | Versata | |
| Macromedia | | |
| Merant | | |
| Mid-Comp | | |



www.Eclipse.org

"This has long been a dream of developers, and now IBM is providing the integration framework to make it possible."
Arun Gupta, CTO, Neuvis Software

"The goal of Eclipse to provide a seamless interface where developers can leverage tools from multiple vendors is a great idea and is something that will add value to our customers. Eclipse is a great way to make it easy for iSeries customers to integrate WebSphere and leading development tools like LANSAS."

Bill Benjamin
Vice President, Business Development
LANSAS Inc.

Not a complete list!!



What is WebSphere Studio Workbench?

WebSphere software

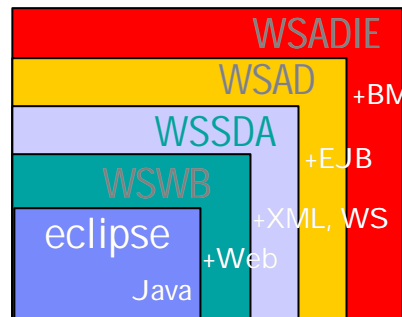
@business on demand software

f WebSphere Studio Workbench is:

- Basis of all next-gen IBM AD products. Eg:
 - f* WebSphere Studio Site Developer
 - f* WebSphere Studio Application Developer
 - f* WebSphere Studio Integration Edition
- Available to IBM Business Partners
 - f* Via IBM PartnerWorld for Developers
 - ✓ To build/test plugins
 - ✓ To create products

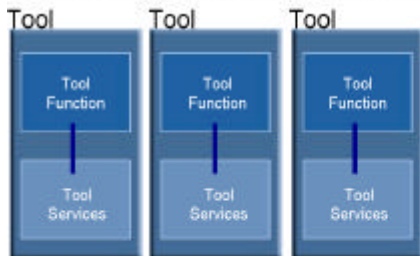
Each product is a different "configuration" of WebSphere Studio

f Refers to product family



WebSphere Studio Workbench (WSWB)

A fundamental change in AD Tooling paradigm

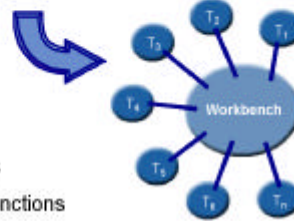


Challenges

- ▶ Difficulty of integration
- ▶ Different Semantics, UI etc
- ▶ Different repositories
- ▶ Not best-of-breed Tool Services
- ▶ Difficult to manage
- ▶ Slow to Market

Benefits

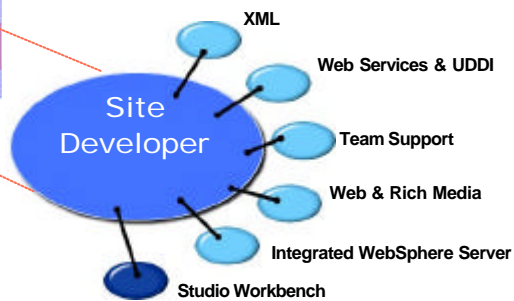
- ▶ Easier integration
- ▶ Single view/mgmt
- ▶ Common look and feel
- ▶ Best-of-breed Tool Services
- ▶ Tool builder focus on tool functions
- ▶ Fast to Market



The New WebSphere Studio

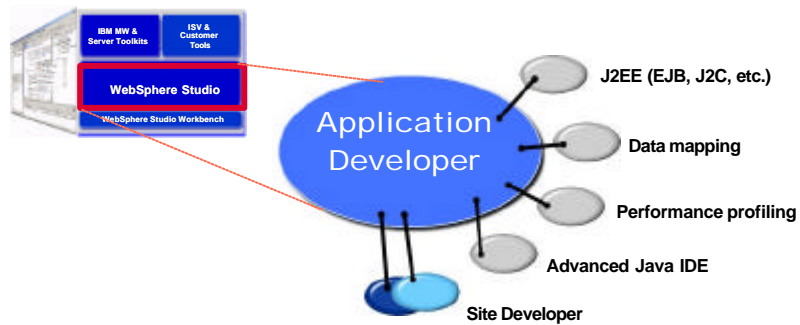
Site Developer

- ★ For professional developers of dynamic Web applications & sites
- ★ Delivers integrated support for open Web standards, including Java, JSP, XML, Rich Media & Web services tools



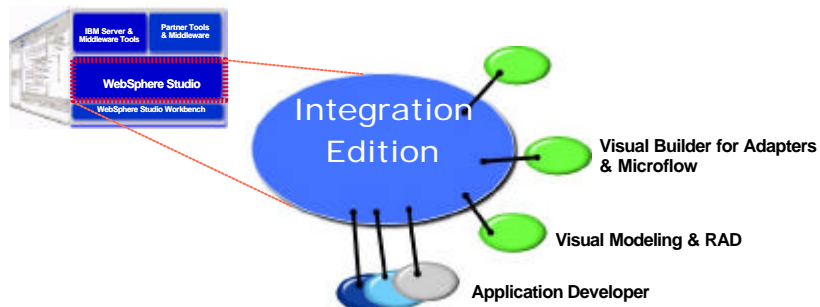
The New WebSphere Studio Application Developer

- ★ For professional developers of Java & J2EE applications, requiring integrated Web, JSP, XML & Web services support
- ★ Advanced tools for code generation, performance tuning & quality
- ★ Includes a highly integrated WebSphere Application Server test environment and deployment automation tools



The New WebSphere Studio AD Integration Edition

- ★ For developers & integrators of advanced J2EE & Web services
- ★ RAD tools for visual application modeling, composition & adaptability (e.g., business rules)
- ★ Advanced EAI tools for generating adapters & dynamic microflows including integration of remote host assets



Eclipse Technology and Terminology

WebSphere software

@business on demand software

Workbench



- Eclipse UI Workbench
 - ▶ Perspectives
 - Contain views and editors
 - Are named and have default layout of the views/editors
 - Are targeted towards a specific type of development, i.e. Java, XML, Help, Debug, WebFacing, ...
 - ▶ Editors
 - That open, edit, save and close files
 - Can contribute to the Workbench menu bar and tool bar
 - ▶ Views
 - That display information
 - Can have their own toolbars
 - Updates saved automatically
 - ▶ Miscellaneous
 - Actions, Menus, Toolbars, Wizards, Properties, Preferences

Workspace

- Workspace is the term given to refer to all development resources currently accessible in the Workbench
- Eclipse Resource Workspace
 - ▶ Resources = Projects + Folders + Files
 - Projects map to a folder in local file system
 - Projects contain folders and files
 - Folders contain folders and files
 - Projects are typed
 - ▶ Resources are versioned as they change
 - User can compare and restore versions
 - ▶ Resources can have properties
 - Properties are displayed in the properties view



Workbench Overview

The screenshot shows the Eclipse IDE interface with several callout boxes pointing to specific features:

- Global tool bar and menu bar:** Points to the top menu bar (File, Edit, Project, Run, Window, Help) and the toolbar below it.
- Open perspectives:** Points to the perspective icon on the left side of the IDE.
- Pull down menu for a view:** Points to a dropdown arrow on the left side of the IDE.
- Resources:** Points to the Outline view on the left, showing a tree structure of the project.
- Editor:** Points to the main central area where the XML file is being edited.
- Local tool bar for a view:** Points to a toolbar located below the Outline view.
- Views:** Points to the bottom of the IDE, specifically the 'Tasks' and 'Properties' tabs.

The main editor displays an XML document with the following content:

```

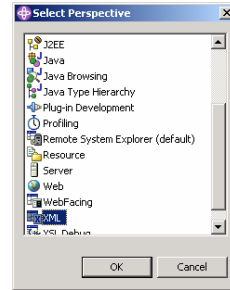
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<purchaseOrder xmlns:po="http://www.ibm.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ibm.com PurchaseOrder.xsd">
  <shipTo>
    <country>US</country>
    <name>Alice Smith</name>
    <street>125 Maple Street</street>
    <city>Mill Valley</city>
    <state>CA</state>
    <zip>90952</zip>
  </shipTo>
  <billTo>
    <po:comment>Hurry, my</po:comment>
  </billTo>
  <items>
    <item>
      <productName>partNum</productName>
    </item>
  </items>
</purchaseOrder>
    
```

The Properties view at the bottom shows the following data:

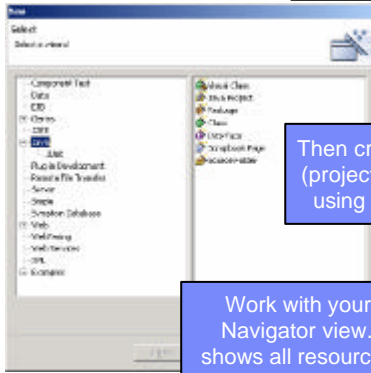
Property	Value
editable	true
last modified	20/05/03 6:34 PM
name	PurchaseOrder.xml
path	/XMLEditProject/PurchaseOrder/PurchaseOrder.xml
size	1061

Using Eclipse

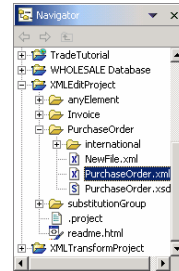
Start by opening a perspective based on the type of development they are working on.



Then create new resources (projects, files and folders) using the "New" wizards



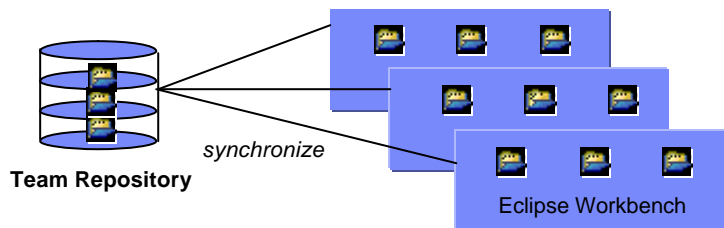
Work with your resources using the Navigator view. The Navigator view shows all resources. Often it is easier to use a local view which filters out projects that are not related. For example the "Packages view" shows only Java projects.



Eclipse Team Support

■ Team

- ▶ Users create and work with **projects**
 - Contain **folders** and **files**
 - Which are versioned as they change
 - Are "typed" and have type-unique tooling
 - Are **synchronized** between local workspace and team repository



Developing Eclipse Plug-ins

WebSphere software

@business on demand software

Plug-in Development

- You add functions to the Workbench by developing plug-ins
- Plug-ins
 - ▶ Highest level of organization / structure for adding functionality to the Workbench
 - WebFacing plug-in
 - Remote Systems Explorer plug-in
 - Web Tooling plug-in
 - iSeries extensions to Web Tooling plug-in
 - ...
 - ▶ A collection of files which implement your desired function
 - ▶ Plugin installed by copying your files to a subdirectory of <root>/plugins
 - There are built in tools for installing new plug-ins
 - ▶ Described by the plugin.xml file
 - Eclipse will look for this file in all subdirectories of <root>/plugins
 - Describes what function(s) your plug-in is adding to the Workbench

Anatomy of a Plug-in

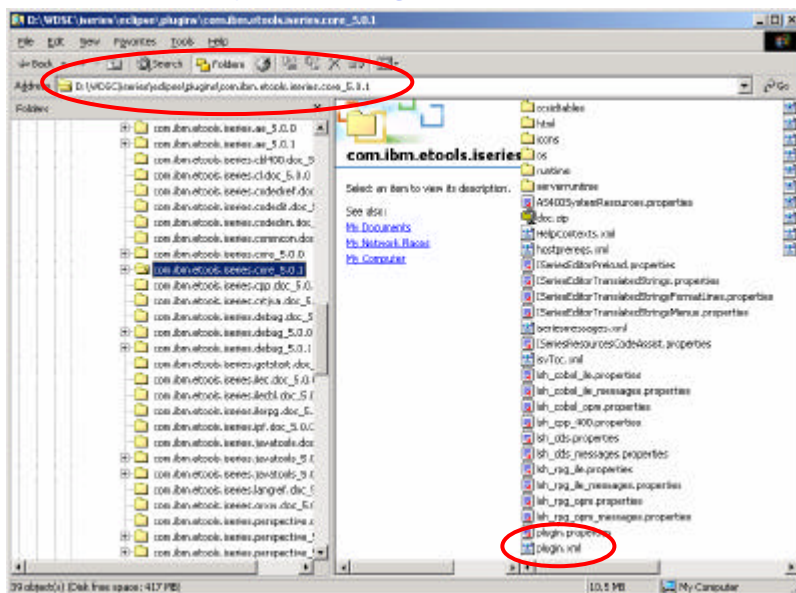
- Files typically found in a plug-in

```
<root>/plugins/com.ibm.example/
  plugin.xml
  plugin.properties
  example.jar
  example.properties
  HelpContexts.xml
  doc.properties
  doc/
    *.html
  n/
    en_US/
      ...
  icons/
    ...
```

- describes the plug-in
- translatable strings for plugin.xml
- Java code for the plug-in
- translatable strings
- structure for F1 help
- translatable strings for help
- plug-in documentation
- NL versions of .properties files
- graphics files for your plug-in



Example Anatomy of a Plug-in



Plugin.xml File

- Plugin.xml
 - ▶ An XML file that describes:
 - Prerequisite plug-ins
 - Jar files shipped with your plug-in
 - Extension points implemented by your plug-in
 - Declares new extension points added by your plug-in
 - ▶ Required for every plug-in
 - ▶ Special plugin.xml editor included in the Eclipse plug-in development environment

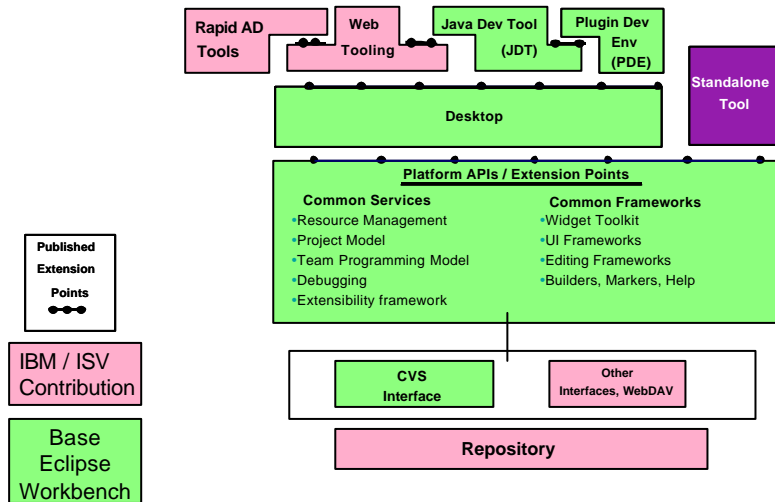


Extension Points

- Extension points allow you to add functions to the workbench
 - ▶ Declared in the plugin.xml file
 - ▶ Associated Java code is shipped in JAR files
 - ▶ First you need to find the extension point for the new function you want to add
 - Workbench extension points are documented in the online help
 - Includes the XML DTD for the extension point and example XML code



Extension Points



Example Extension Points

Help - WebSphere Development Studio Client Advanced Edition for iSeries

Search: [Advanced Search](#)

Contents

Tool developer information

- Tool developer information
 - Extending the platform
 - Programmer's Guide
 - Reference
 - API Reference
 - Extension Points Reference
 - Platform Runtime
 - Workspace
 - Workbench
 - Team
 - Debug
 - Help
 - Other
 - Other Reference Information
 - Examples Guide
 - Questions Index
 - Extending the Java tools
 - Plug-in Development Environment
 - Help System Example
 - GEF Developer's Guide
 - Draw2D Developer's Guide

Workbench

 - [org.eclipse.ui.acceleratorConfigurations](#)
 - [org.eclipse.ui.acceleratorScopes](#)
 - [org.eclipse.ui.acceleratorSets](#)
 - [org.eclipse.ui.actionDefinitions](#)
 - [org.eclipse.ui.actionSetPartAssociations](#)
 - [org.eclipse.ui.actionSets](#)
 - [org.eclipse.ui.decorators](#)
 - [org.eclipse.ui.dropActions](#)
 - [org.eclipse.ui.editorActions](#)
 - [org.eclipse.ui.editors](#)
 - [org.eclipse.ui.elementFactories](#)
 - [org.eclipse.ui.exportWizards](#)
 - [org.eclipse.ui.importWizards](#)
 - [org.eclipse.ui.markerHelp](#)
 - [org.eclipse.ui.markerResolution](#)
 - [org.eclipse.ui.newWizards](#)
 - [org.eclipse.ui.perspectiveExtensions](#)
 - [org.eclipse.ui.perspectives](#)
 - [org.eclipse.ui.popupMenus](#)
 - [org.eclipse.ui.preferencePages](#)
 - [org.eclipse.ui.projectNatureImages](#)
 - [org.eclipse.ui.propertyPages](#)
 - [org.eclipse.ui.resourceFilters](#)
 - [org.eclipse.ui.startup](#)
 - [org.eclipse.ui.viewActions](#)
 - [org.eclipse.ui.views](#)
 - [org.eclipse.ui.workingSets](#)

Team

[org.eclipse.team.com.flatunc](#)

Documentation for Workbench extension points is available in the online help under *Tool Developer Information*

Add your own popup menus to the Workbench

Extension Point Documentation

Export Wizards

Identifier: org.eclipse.ui.exportWizards

Description: This extension point is used to register export wizard wizards appear as choices within the "Export Dialog", and are used from the workbench.

Wizards may optionally specify a description subelement whose body text about the wizard.

Configuration Markup:

```
<!ELEMENT wizard (description? selection*)>
<!ATTLIST wizard
  id CDATA #REQUIRED
  name CDATA #REQUIRED
  class CDATA #REQUIRED
  icon CDATA #IMPLIED
>
<!ELEMENT description (#PCDATA)>
```

- **id** - a unique name that will be used to identify this wizard
- **name** - a translatable name that will be used in the dialog box to represent the wizard.
- **class** - a fully qualified name of the class that implements

DTD description for the XML to declare the extension point in your plugin.xml file

Actual XML examples are shown farther down the page!

Hello World Plug-in

- Now lets walk through an example
 - ▶ Create a new plug-in project
 - ▶ Examine the Plug-in Development perspective
 - ▶ Add a menu and menu item to the main menu
 - ▶ Supply an action for the menu item that displays a message box

Creating a new Plug-in Project

New
Select
Create a Plug-in Project

Component Test
Data
EJB
iSeries
J2EE
Java
Plug-in Development
Remote File Transfer
Server
Simple
Symptom Database
Web
WebFacing
Web Services
XML
Examples

Plug-in Project
Fragment Project
Extension Point Schema
Convert Java to Plug-in Projects
Feature Project

Use the "New" wizard to create a new Plug-in project

< Back Next > Finish Cancel

New Plug-in Project
Plug-in Project Name
Select the name of the new plug-in project (e.g. 'com.example.xyz'). By default, this name will also be used for the plug-in ID.

Project name: com.example.MyPlugin

Project contents:
 Use default

Directory: C:\Documents and Settings\jvantsa\My Documents\IBM\wdsc\workspace Browse...

By convention your plug-ins should start with the reverse of your Internet domain name. For example: com.ibm...

< Back Next > Finish Cancel

Creating a New Plug-in Project

New Plug-in Project
Plug-in Project Structure
Define essential plug-in structure settings

Plug-in ID: com.example.MyPlugin

Java builder output: bin

Plug-in runtime library: MyPlugin.jar

Source folder: src

< Back Next > Finish Cancel

New Plug-in Project
Plug-in Code Generators
Select the wizard that will generate the initial plug-in code

Create a blank plug-in project
 Create a plug-in project using a code generation wizard

Available code generation wizards

- Custom plug-in wizard
- Default Plug-In Structure
- Hello, World
- Plug-in with a multi-page editor
- Plug-in with an editor
- Plug-in with a popup menu
- Plug-in with a property page
- Plug-in with a view
- Plug-in with perspective extensions

This wizard creates standard plug-in directory structure and adds the following:

- Action set.** This template creates a simple action set that adds **Sample Menu** to the menu bar and a button to the tool bar. Both the menu item in the new menu and the button invoke the same **Sample Action**. Its role is to open a simple message dialog with a message of your choice.

Extensions Used

- org.eclipse.ui.actionSet

< Back Next > Finish Cancel

Creating a New Plug-in Project

New Hello World plug-in project

Simple Plug-in Content
Enter the required data to generate initial plug-in files.

Plug-in name:

Version:

Provider Name:

Class Name:

Generate code for the class

Plug-in code generation options

Add default instance access

Add support for resource bundles

Add access to the workspace

New Hello World plug-in project

Sample Action Set
This template will generate a sample action set extension with a menu, a menu item and a tool bar button.

Java Package Name:

Action Class Name:

Message Box Text:

Add the action set to the resource perspective



Plug-in Development Environment

Plugin project (points to Package Explorer)

plugin.xml Editor (points to the main editor area)

Plug-in development perspective (points to the right-hand side of the IDE)

Implementation of the actionSets extension point to provide the popup menu (points to the Extensions section in the editor)

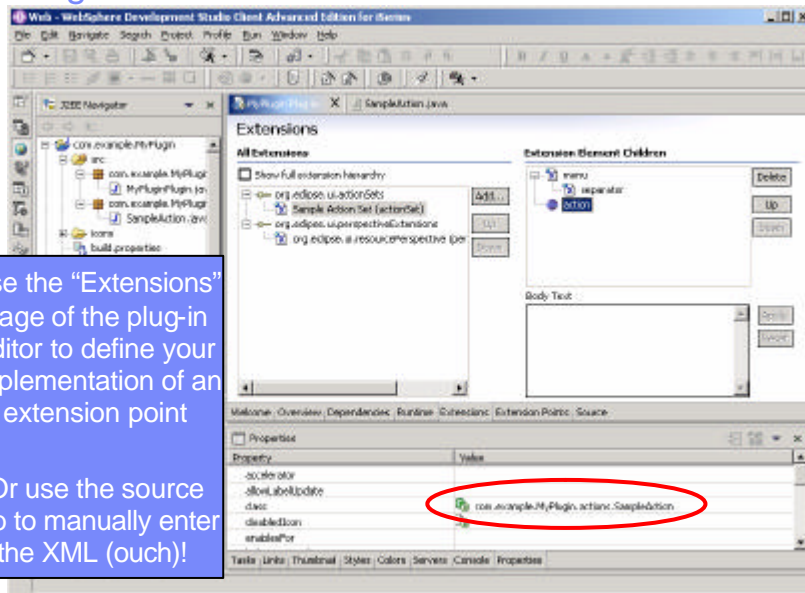
Customized editor pages to help you define extensions, runtime jar files, dependencies, ... (points to the bottom tabs of the editor)



Defining the Extension

Use the "Extensions" page of the plug-in editor to define your implementation of an extension point

Or use the source tab to manually enter the XML (ouch!)

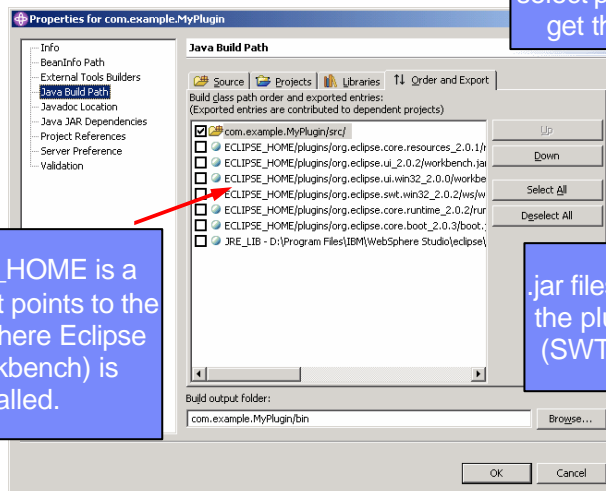


Plug-in Build Path

Right click on the PDE project and select properties to get this dialog

ECLIPSE_HOME is a variable that points to the directory where Eclipse (the workbench) is installed.

.jar files required for the plug-in to build (SWT, JFace, ...)



Action Class (Invoked by the Popup Menu)

actionSet extension point requires the class to implement the `WorkbenchWindowActionDelegate` interface

```

public class SampleAction implements IWorkbenchWindowActionDelegate {
    private IWorkbenchWindowActionDelegate delegate;

    public SampleAction(IWorkbenchWindowActionDelegate delegate) {
        this.delegate = delegate;
    }

    public void run(IAction action) {
        MessageDialog.openInformation(
            window.getShell(),
            "Hello from Plug-in",
            "Hello, Eclipse world!");
    }
}
    
```

JFace MessageDialog class

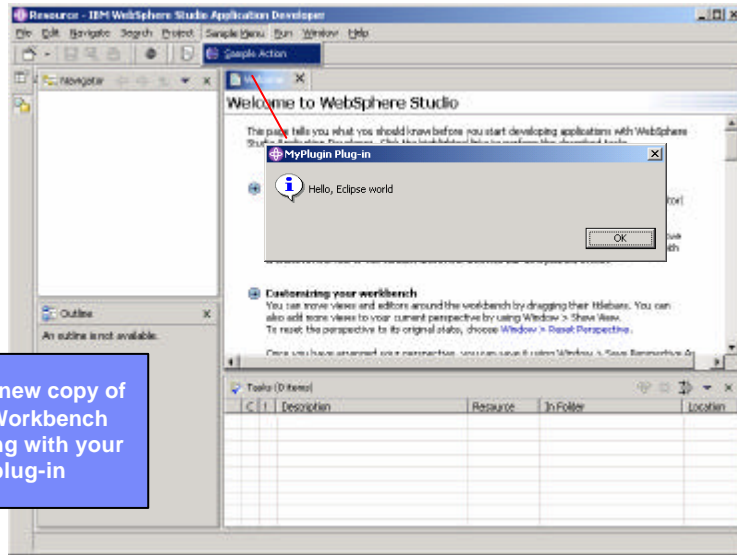
run method(...) is invoked when the popup menu is selected

Testing Your Plug-in

1. Select any plug-in project
2. Use the Run As -> Run-time Workbench menu item to launch a new instance of the Workbench with your plug-in(s) loaded.

- f Use Debug-> Run-time Workbench to debug your plug-in
- f Set breakpoints in the Java source editor before or after launching the debugger

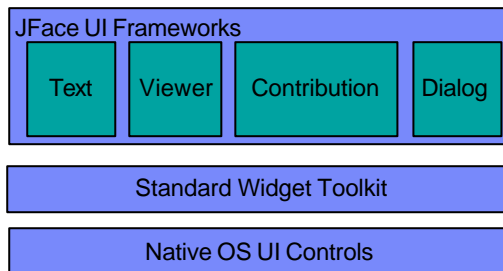
It Works!



Brand new copy of the Workbench running with your plug-in

User Interface

- User Interface for plug-ins should be done using SWT and JFace
 - ▶ Provides the best integration with the Workbench
- SWT is a new Java widget toolkit
 - ▶ Different than AWT and Swing
- JFace is a UI Framework built on top of SWT for creating wizards, editors, views, dialogs, preference pages,



User Interface

- SWT
 - ▶ Fast
 - ▶ Native operating system look and feel
 - ▶ Provides all the common UI widgets
 - Button, Label, Menu, MenuItem, ProgressBar, ...
- JFace
 - ▶ UI frameworks built on top of SWT
 - Provide a higher level of abstraction so you can focus on your added function, not UI details
 - ▶ Provide a common look and feel across all Workbench tools
 - Wizards, ..., preference pages all look and act the same
 - Provides base set of classes you can extend to add your specific function



User Interface

- JFace
 - ▶ Text Framework
 - Framework for providing source (text) editors
 - Provides support for
 - Content assist, syntax highlighting, formatting, ...
 - ▶ Viewer Framework
 - Different types of views into a data model
 - TreeViewer, CheckboxTreeViewer, TableViewer, ...
 - ▶ Contribution Framework
 - Framework for adding actions to the workbench toolbar and menubar
 - ▶ Dialog Framework
 - Framework for creating dialogs that interact with the user
 - Example classes:
 - PreferenceDialog, PropertyDialog, WizardDialog, ...
 - Also provides pages for using inside the dialogs
 - PreferencePage, PropertyPage, WizardPage, ...



Other Things to Consider

WebSphere software

@business on demand software

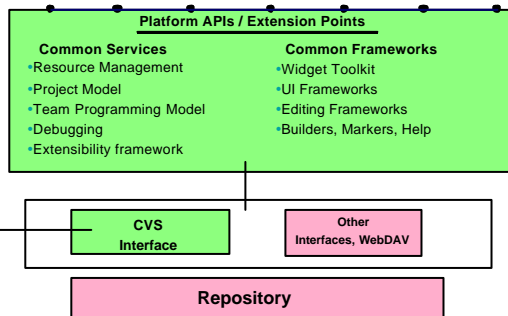
HELP!!!

- Lots of ways to provide help!
 - ▶ Tooltip help
 - Also known as "fly-over help"
 - Tooltip is shown when user hovers the mouse pointer over a menu item, button, entry field, ...
 - Use the `setToolTipText(String text)` method on a widget
 - ▶ F1 key help
 - Context sensitive help
 - Typically used to provide a short description of the current task when the user presses F1 in a view, dialog, wizard, entry field
 - Set context ids using `WorkbenchHelp.setHelp(...)`
 - Eclipse then looks up the context id in `HelpContexts.xml`
 - ▶ Full blown online help
 - You can provide complete online documentation integrated into the *Help* perspective
 - This is full topic on its own

Team Extension Points

- Version / Configuration Management
 - ▶ Allows source management providers to plug into the workbench
 - ▶ Can be used with by any project in the Workspace without any additional work
 - Java projects, Web projects, iSeries projects, ...

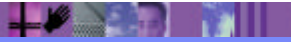
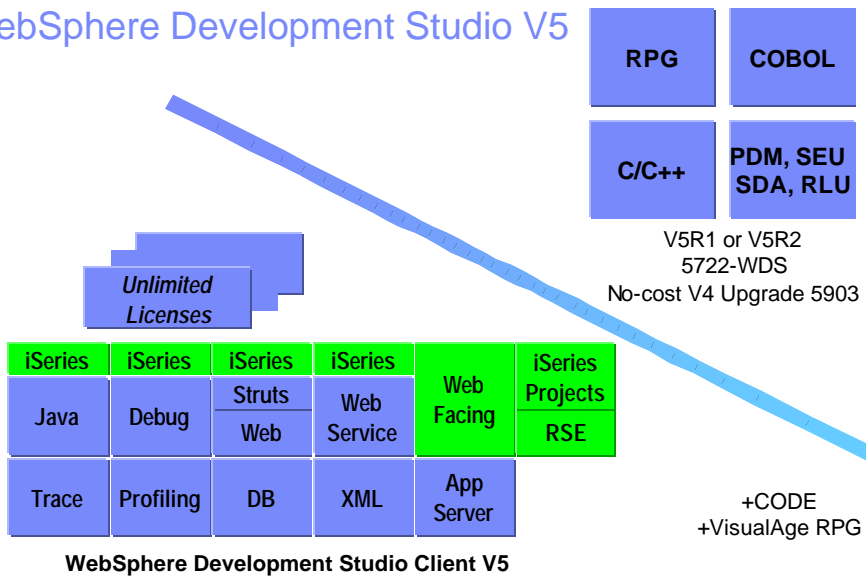
Open source CVS implementation of the VCM APIs is provided as a reference implementation



What About the iSeries?

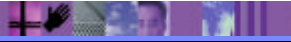


WebSphere Development Studio V5



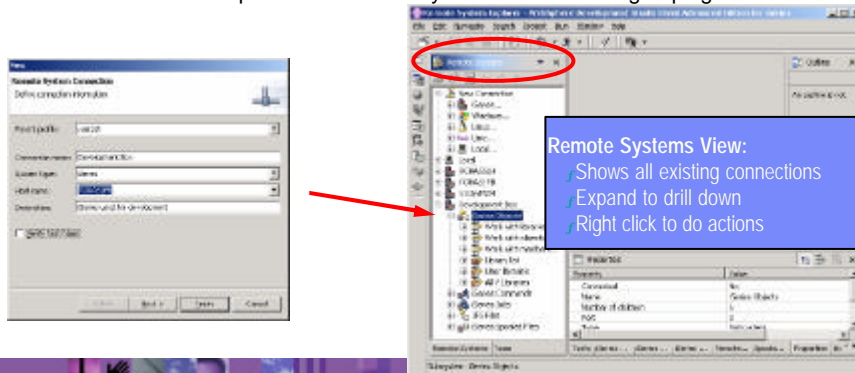
WDSc: RPG and COBOL Tools

- iSeries Tools for RPG and COBOL
 - ▶ **Remote Systems Explorer**
 - For random drill-down or filtered access to remote:
 - Library, objects and members in QSYS
 - IFS Folders and Files
 - Linux (including LPAR), Unix and Windows Folders and Files
 - Local Folders and Files
 - For remote command execution of
 - QSYS Commands
 - QShell Commands
 - Linux, Unix and Windows Commands
 - For exploration of Jobs in QSYS
 - ▶ **iSeries Projects and Navigator**
 - For structured, project-based development
 - Local team-sharable project
 - Holds source files and source members and save files
 - Push action to copy local files to associated library



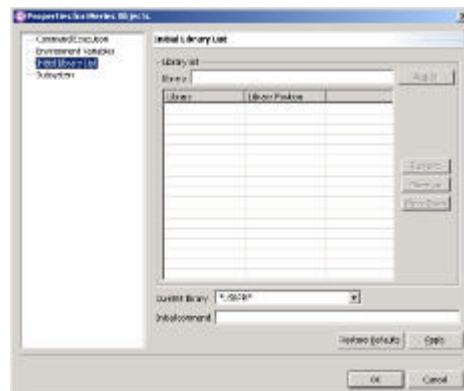
Remote System Explorer (RSE)

- Remote Systems Explorer Perspective
 - **Manages connections to remote systems**
 - Where you create and manage connection information
 - Connections subsequently used in all iSeries tools
 - Start by using the New Connection wizard
 - Connection appears in tree view
 - Expand it to see "subsystem" -> functional groupings



iSeries Connection Properties

- iSeries Connection Properties
 - Use Properties of iSeries Commands to set:
 - Library list information
 - Current library
 - Initial command to call
 - Environment variables
 - All information is applied at connect time



iSeries Connection Subsystems

- Expand a connection in the RSE to see subsystems:
 - ▶ **iSeries Objects**
 - Like WRKXXXXPDM. For drilling down on libraries, objects, mbrs
 - Right click offers wealth of actions like create, rename, delete...
 - Can edit source members using builtin LPEX or classic CODE
 - Can create User Defined Actions just like in PDM
 - ▶ **iSeries Commands**
 - For creating command sets containing one or more CL commands
 - Right click and select Run to remote run commands
 - Results are logged in Command console
 - ▶ **iSeries Jobs**
 - For seeing jobs, filtered by number, user ID or status
 - Right click to do common actions on jobs, such as end or hold
 - ▶ **IFS Files**
 - For drilling down on IFS folders and files
 - Right click offers wealth of actions like create, rename, delete...
 - Can edit source members using builtin LPEX or classic CODE

IBM Software Group | WebSphere software

•Expand a filter to see objects resulting from resolution of filter

•Drill down through libraries, and source files to get to members

•Right click on libraries, objects and members to get actions

Built-in Remote Systems LPEX Editor

Various iSeries specific views for showing additional information. iSeries Error List shows compile / verify errors

RSE: Filters and Filter Pools

- Expand a subsystem to see "filters"
 - ▶ Like WRKXXXPDM generic name filters but
 - Are named and saved
 - ▶ There are some predefined
 - Such as Library List to see *LIBL libraries in iSeries Objects
 - Such as popular commands in iSeries Commands
 - ▶ To create your own, right click on subsystem object
 - ▶ Each filter can contain multiple filter strings
 - So you can list all libraries that start with A and B
- Eventually, you'll have too many filters
 - ▶ Time to turn on "Show Filter Pools" via preferences
 - Expanding subsystems will then first show filter pools
 - ▶ Filters are grouped into named pools
 - By default they are added to the single default filter pool
 - ▶ You can create your own filter pools
 - Then add filters to it
 - ▶ Expand a filter pool to see just the filters in it



iSeries Objects Subsystem Filters

- **Library List filter**
 - ▶ Displays library list
 - ▶ Provides popup actions for working with the library list
- **Library filters**
 - ▶ Specify simple, generic or special library names
 - Lists all matching libraries, when expanded
- **Object filters**
 - ▶ Specify simple or generic object names, library-qualified
 - Library name can be simple, generic or special
 - ▶ Specify simple or generic object types and attributes
 - Can specify one or more type:attribute pairs (OR operation)
 - Lists all objects that meet criteria, when expanded
- **Member filters**
 - ▶ Specify simple or generic member names, lib/file-qualified
 - Library name and file name can be simple or generic
 - ▶ Specify simple or generic member types
 - Can specify one or more member types (OR operation)
 - Lists all members that meet criteria, when expanded

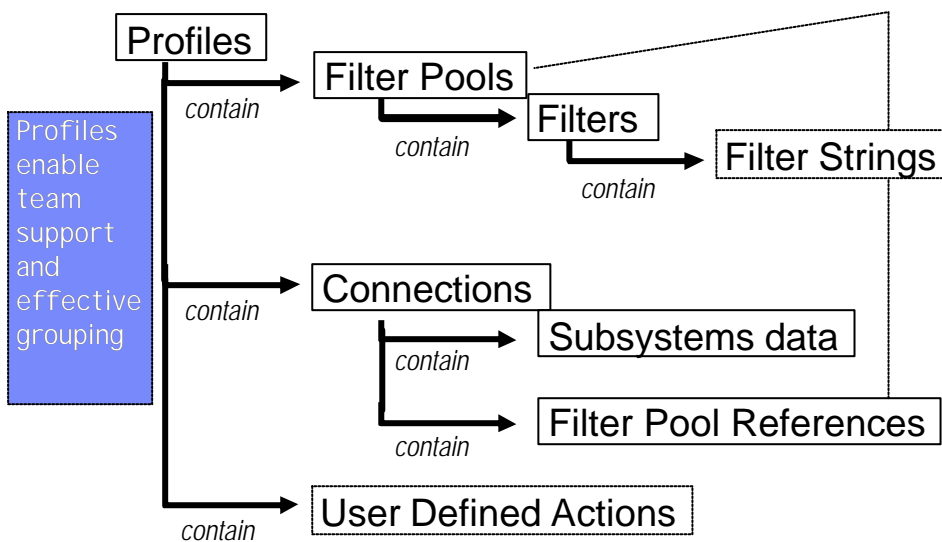


Remote Systems LPEX Editor

- Remote Systems LPEX Editor
 - ▶ Re-write of CODE Editor for Eclipse
- What it has today:
 - ▶ All base support from CODE Editor, such as:
 - Alt+L/C/M/D/U/S/J to select/copy/move/delete/unselect/split/join
 - Select streams, lines, blocks and rectangles (Edit menu)
 - Command-line for directly entering editor commands
 - Keystroke recording for repetitive tasks
 - Compare utility
 - Rich search, find and replace
 - Bookmarks and quick marks
 - Ctrl+L to locate a line
 - Line-number prefix area
 - File->Get File to import a file
 - Filter Selection to only show lines containing selected text
 - ▶ Plus!
 - Outline view support for ILE RPG and COBOL
 - Content assist for ILE RPG



Underlying Model of the RSE



iSeries Projects Perspective

Callout boxes in the screenshot:

- Create iSeries Project
- Create Source File
- Create Source Mbr
- iSeries Project
- Source Physical File
- Source Members
- iSeries Project navigator: expand project to work with local source
- Remote Systems view for convenience

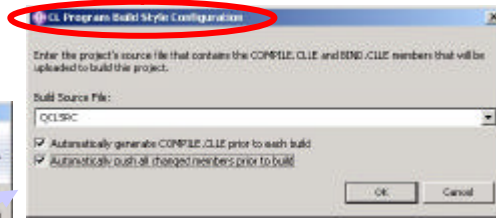
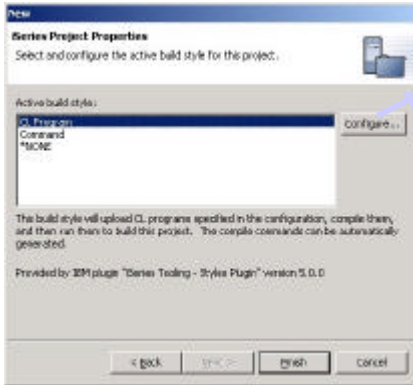
New iSeries Project Wizard

Callout boxes in the screenshots:

- Give your project a name
- Choose connection from RSE list
- Identify Associated Library: this is the target library for pushes and builds

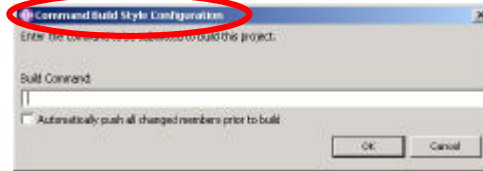
iSeries Project Wizard

Build style controls how the iSeries project is built on the remote system



Specify Build Style and style configuration parameters for iSeries Project

more information on next slide ...

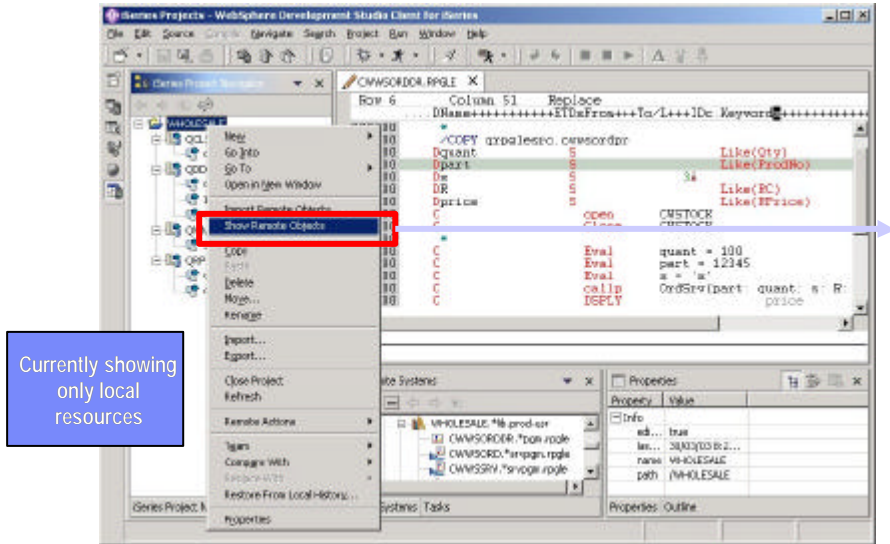


iSeries Projects Build Styles

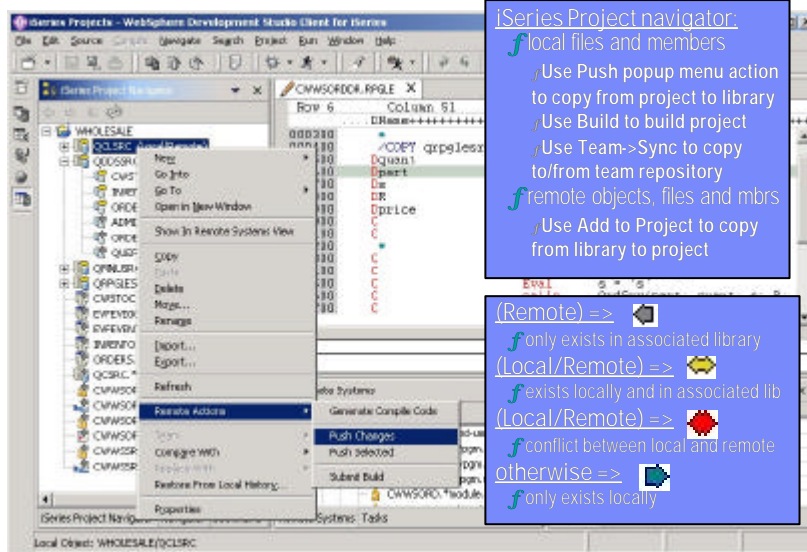
- Build Styles
 - ▶ Specify how to build the iSeries project on the remote system
 - ▶ Three IBM-supplied build styles
 - ▶ **CL Program**
 - Automatically generates COMPILE.CLLE and BIND.CLLE
 - Adds compile command for each source file in project
 - Uses last compile command from RSE for specified source type
 - i.e. CRTBNDRPG versus CRTRPGMOD
 - At build time these two members are pushed to iSeries, compiled and run
 - ▶ **Command**
 - You specify single command to run build
 - Good if you have an existing build script
 - ▶ ***NONE**
 - No build support
- ISVs can plug-in additional build styles
 - ▶ To work with their existing iSeries systems



Project "Bleedthrough"



Local and Remote Objects



iSeries Projects Remote Build

- **Submit Build**
 - ▶ Submits build of iSeries project on remote system
 - ▶ Adds build job to iSeries Build Job Status view

The screenshot shows a table titled "iSeries Build Job Status" with columns for Date/Time Submitted, Job, Project, and Status. The table contains two rows of data. A context menu is open over the second row, showing options like "Retrieve Error" and "Retrieve".

| Date/Time Submitted | Job | Project | Status |
|----------------------|----------------------|-----------|----------|
| 30-Nov-83 9:29:02 PM | 823621(YANT21)YANT21 | MAIN PROJ | Finished |
| 30-Nov-83 9:30:06 PM | 823622(YANT21)YANT21 | MAIN PROJ | Queued |

Callouts in the image explain the following elements:

- Job where build is running:** Points to the Job column.
- Status of the build:** Points to the Status column.
- Refresh interval determines how frequently status is updated:** Points to the "Refresh interval: 10 seconds" text at the bottom of the window.
- Retrieve errors from build to the iSeries Error List:** Points to the "Retrieve Error" option in the context menu.

Remote System Explorer Extension Points

Additional documentation on the following extension points can be found in:

C:\WDSC\iseries\eclipse\plugins\com.ibm.etools.systems.core_5.0.0\plugin.xml

Benefits of Building On Top of the RSE

- What does the RSE offer you?
 - ▶ **Extension points for remote objects**
 - Popup menus
 - Property pages
 - ▶ **Extend the remote system framework**
 - Add a new subsystem
 - Add a new system type
 - iSeries, Linux, Unix, Windows and Local are pre-supplied
 - ▶ **Java APIs for accessing remote systems**
 - Native
 - Retrieve lists of libraries, objects, members and jobs based on filters
 - Retrieve properties for a library, object, member or job
 - Prompt and run commands
 - Run applications in batch or interactive
 - IFS / Windows / Linux / Unix
 - Retrieve lists of directories / files
 - Run remote commands



Adding a New RSE System Type

- **Extension point**
 - ▶ com.ibm.etools.systems.core.systemtype
- **Adds a new system type to the Remote Systems Framework**
 - ▶ The new system type will appear in the new connection wizard
 - ▶ No Java code required!
 - ▶ You can then add new subsystems for this type

```
<extension point="com.ibm.etools.systems.core.systemtype">
  <type name="Windows"
    iconlive="icons/windowsLive.gif">
    icon="icons/windows.gif">
  </type>
</extension>
```

icon changes
when system
is connected
or
disconnected



Adding New Subsystems

- **Extension point**

- ▶ com.ibm.etools.systems.core.subsystemfactory
 - A SubSystemFactory creates instances of a specific type of subsystem

- **Adds a new subsystem**

- ▶ New subsystem will appear under connections of specified type

```
<extension point="com.ibm.etools.systems.core.subsystemfactory">
  <factory
    systemtypes="iSeries"
    name="Files"
    icon="icons/filessubsyst.gif"
    iconlive="icons/filessubsystlive.gif"
    id="com.acme.etools.mypkg.myfactory"
    class="com.acme.etools.mypkg.MyClass"
    category="files"
    vendor="acme">
  </factory>
</extension>
```

Class must implement the
com.ibm.etools.systems.subsystems.SubSystemFactory
interface



Adding Property Pages for Remote Objects

- **Extension point**

- ▶ com.ibm.etools.systems.core.propertyPages

- **Add property pages for remote objects**

- ▶ Property pages will appear in the properties dialog when user selects "Properties..." from the popup menu for a remote object

```
<extension point="com.ibm.etools.systems.core.propertyPages">
  <page class="com.acme.myPropertyPage"
    id="com.acme.mypropertypage"
    name="Remote File Info"
    icon="icons/myProperties.gif"
    subsystemfactoryid="ibm.filesLocal"
    typecategoryfilter="files"
    typefilter="file"
    subtypefilter="java"
    namefilter="*.java">
  </page>
</extension>
```

Filters are subsystem specific
Each remote object can have a type, subtype and subtype
For example

- Type: *PGM
- Subtype: RPGLE

Can also filter by name



Adding Popup Menu Actions for Remote Objects

- **Extension point**
 - ▶ com.ibm.etools.systems.core.popupMenus
- **Add action for remote objects**
 - ▶ New action will appear in the popup menu of remote objects in the Remote Systems view

Example 



Adding Popup Menu Actions for Remote Objects

```

<extension point="com.ibm.etools.systems.core.popupMenus">
  <objectContribution
    id="com.acme.displayfile.actions"
    subsystemfactoryid="ibm.files400"
    typecategoryfilter="OBJECT*"
    typefilter="*FILE"
    subtypefilter="DSPF"
    namefilter="*">
    <menu id="com.acme.displayfile.menu"
      label="My Display File Actions">
      <separator name="group1"/>
    </menu>
    <action id="com.acme.action1"
      label="My first action"
      class="com.acme.actions.Action1"
      menubarPath="com.acme.displayfile.menu/group 1"
      enablesFor="1">
    </action>
    <action id="com.acme.action2"
      label="My second action"
      class="com.acme.actions.Action2"
      menubarPath="com.acme.displayfile.menu/group 1"
      enablesFor="+">
    </action>
  </objectContribution>
</extension>

```

Same as property pages

Adds the menu item

Your actions





IBM Software Group

Examples from Business Partners

WebSphere software

@business on demand software

SoftLanding Systems, Inc.

The screenshot shows the 'Remote System Explorer' window in IBM WebSphere Studio. The left pane displays a tree view of remote systems, with 'Your spooled files' expanded to show a file named 'ORDENTR'. A context menu is open over this file, listing actions such as 'Go To', 'Refresh', 'Expand', 'Collapse', 'Resume', 'Delete', 'Open with text editor', 'View', 'Save to file...', 'Change', and 'Move'. The main pane shows a table of spooled file details.

| Command | Issued by | Program | Library | Text 'description' |
|-------------------------------------|-----------|---------|----------|--------------------|
| ORDENTR_941573_YANTZL_QPRT306_1.txt | YANTZL | ORDENTR | DEMODY | *SRCMBRTXT |
| 5722WDS V5R2H0 020719 RN | | | | |
| | | ORDENTR | QRPGLSRC | |
| | | DEMODY | | 37 |

Free Spooled File Plug-in for WDS, includes:

- Spooled files subsystem
- Spool files filtering
- Spooled files table view
- Perform remote actions on spooled files

www.softlanding.com

SoftLanding Systems, Inc.

New
Spooled File Filter
 Create a new filter to list spooled files

User:

Output queue:

Library:

User data:

Form type:

< Back Next > Finish Cancel

Define your own filters for showing spooled files

Your spooled files

| File | Device or Queue | User Data | Status | Total Pages | Current P... | User | Job | Job Number | File Num |
|-----------|-----------------|-----------|--------|-------------|--------------|--------|------------|------------|----------|
| JORDENTR | QGPL/QPRINT | | *READY | 33 | 0 | YANTZI | QPADEV0... | 917371 | 1 |
| OFFLINE1A | QGPL/QPRINT | | *READY | 3 | 0 | YANTZI | QPADEV0... | 917406 | 1 |
| OFFLINE1A | QGPL/QPRINT | | *READY | 3 | 0 | YANTZI | QPADEV0... | 917406 | 2 |
| OFFLINE1A | QGPL/QPRINT | | *READY | 3 | 0 | YANTZI | QPADEV0... | 917406 | 3 |
| OFFLINE1A | QGPL/QPRINT | | *READY | 3 | 0 | | | | |
| OFFLINE1A | QGPL/QPRINT | | *READY | 3 | 0 | | | | |
| OFFLINE1A | QGPL/QPRINT | | *READY | 3 | 0 | | | | |
| OFFLINE1A | QGPL/QPRINT | | *READY | 3 | 0 | | | | |

Tasks | Series Table View | Series Commands Log | Properties | Search | Spooled Files

Spooled files view shows details about each pool file and provides right click popup menu actions

Plug-in Development Summary



Summary

- Eclipse and the WebSphere Studio Workbench provide:
 - ▶ **An integrated development environment**
 - ▶ **Extendable platform**
 - You can focus on your function without worrying about:
 - Resource management
 - Help system
 - Rich set of base classes for quickly developing UI
- WDSc provides:
 - ▶ iSeries tooling plugged into Eclipse
 - ▶ Remote Systems Framework
 - With extension points so you can add functionality to the RSE



Additional Resources

- Eclipse
 - ▶ www.eclipse.org
- WebSphere Development Studio Client
 - ▶ <http://www.ibm.com/software/awdtools/series/>
- IBM's developerWorks
 - ▶ www.ibm.com/developer

