Plug-in Development Tips and Tricks

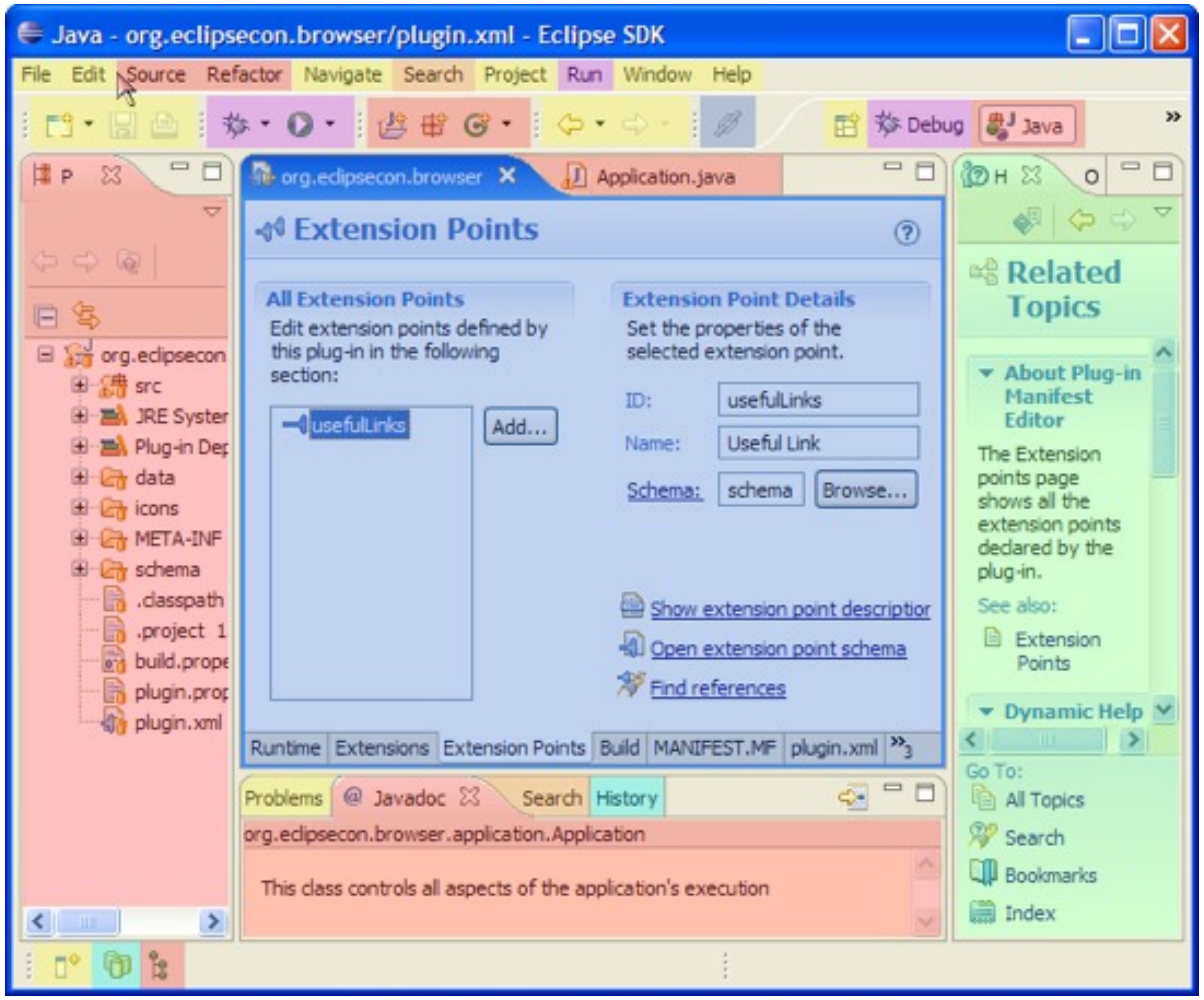
"In PDE we do tooling, but our business is people"

- Chris Aniszczyk <zx@code9.com>
- Principal Consultant at Code 9
- PDE Technical Lead

Agenda

- Plug-in Development with PDE
- Tips and Tricks
- Q&A

Seamless Integration of Components



Edit extension points defined by this plug-in in the following section:

Set the properties of the selected extension point.

ID: usefulLinks

Name: Useful Link

Schema: schema

The Extension points page shows all the extension points declared by the plug-in.

org.eclipsecon.browser.application.Application

This class controls all aspects of the application's execution

Component Legend

🟦	PDE
🟥	JDT
🟩	User Assistance
🟪	Debug
🟨	Workbench
🟧	Search
🟦	Team

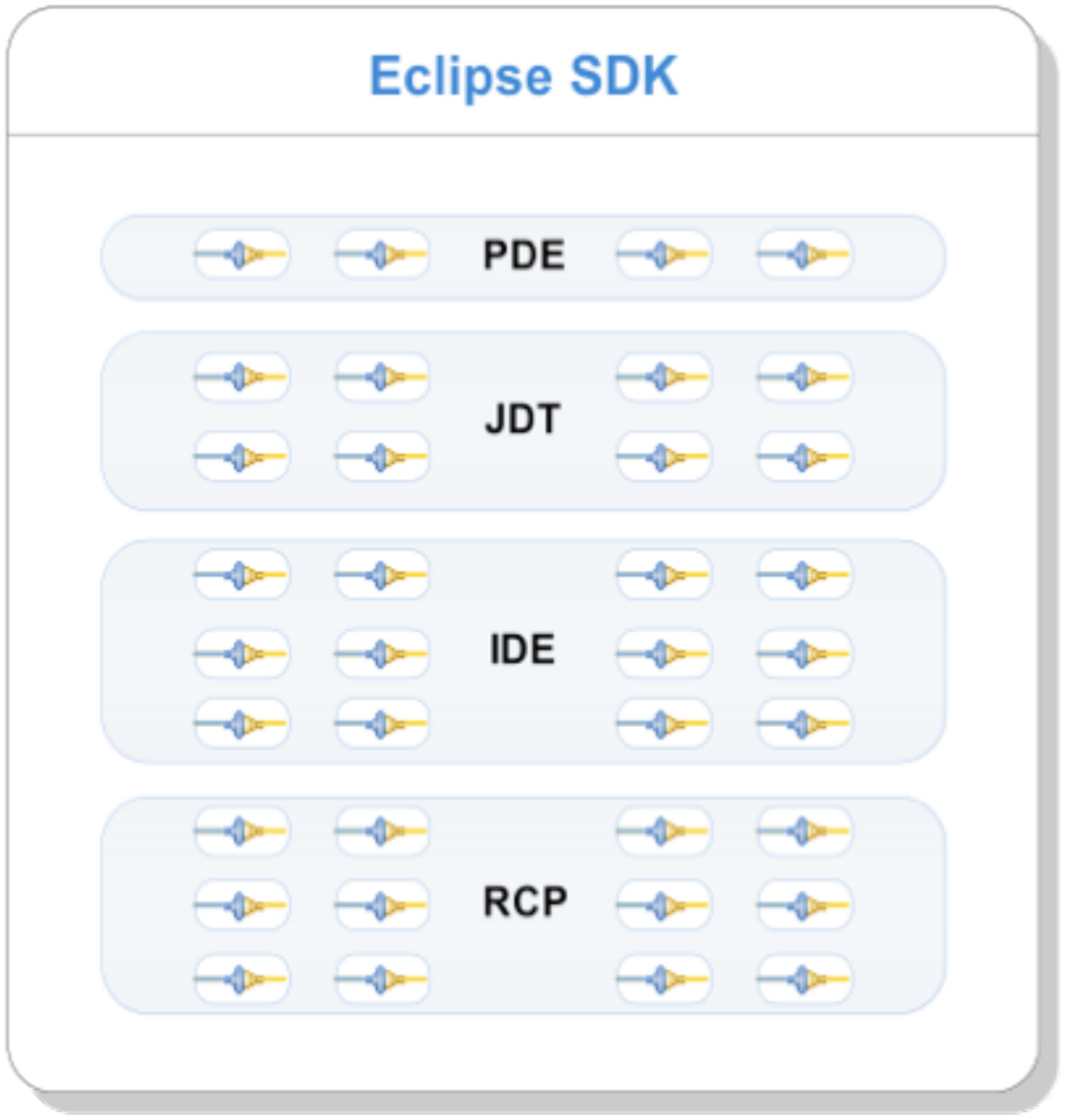
PDE

- PDE = Plug-in Development Environment
- Tools to develop Eclipse plug-ins
  - Wizards to create, import and export plug-ins and features
  - Specialized editors for plug-in manifest files
  - Templates for new plug-ins
  - Launchers to run, debug and test plug-ins
  - NLS tools to internationalize plug-ins
  - Automated class path management

PDE Details

- PDE is implemented as a set of plug-ins
- PDE is built on top of the Eclipse Platform and JDT
  - Uses Eclipse Platform and JDT extension points and APIs
- PDE is seamlessly integrated into Eclipse
- PDE gets no special treatment from the Platform or JDT

Plug-ins All the Way Down



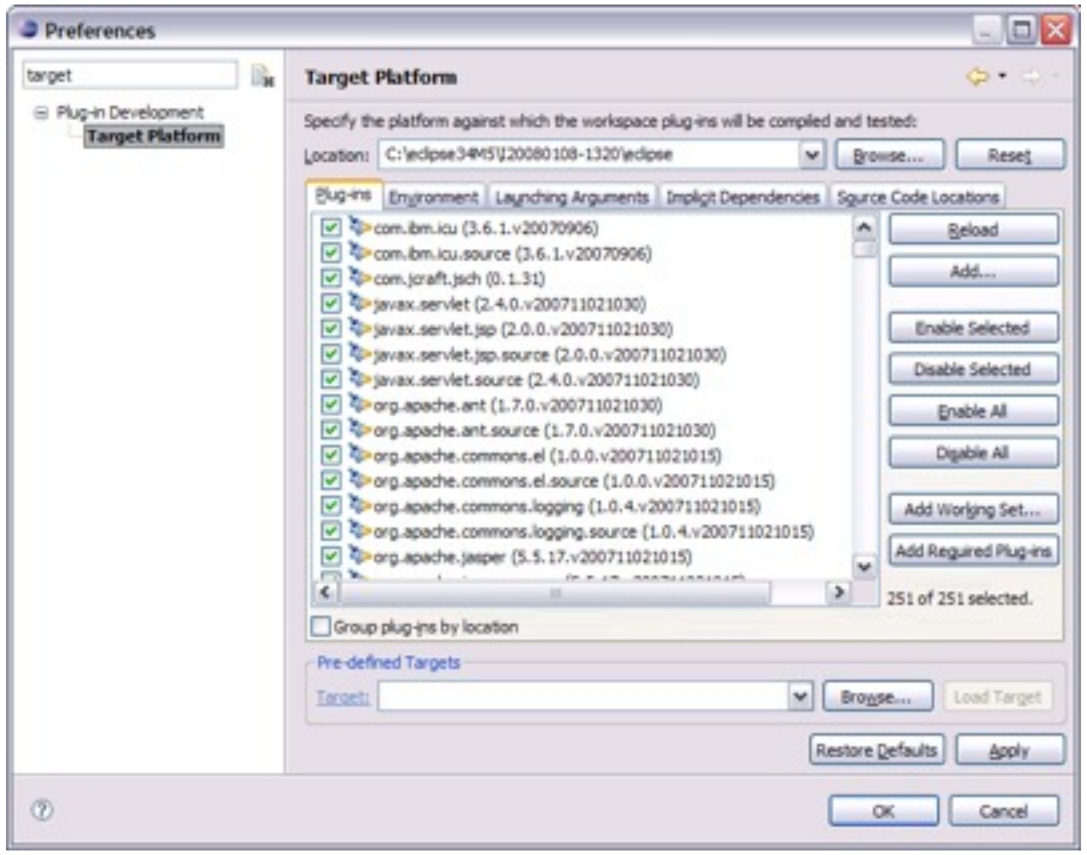
- A plug-in is the fundamental building block of an Eclipse product
- Plug-ins build on top of and use other plug-ins
- To extend Eclipse, you must write plug-ins
- To write a rich client application, you must write plug-ins
- To write an OSGi-based application, you must write plug-ins (bundles)

Agenda

- Plug-in Development with PDE
- Tips and Tricks
- Q&A

Target Management

- Helps you to specify plug-in in which to build and run with.
- Includes tabs to set environment values, launching arguments, implicit dependencies, and source code locations.
- You can add plug-ins to the current target platform by using **target provisioners**. Current provisioners allow you to specify locations on your file system and the locations of update sites.
- The plug-ins can be viewed as a list or a tree (separated by locations).



Target Platform

Specify the platform against which the workspace plug-ins will be compiled and tested:

Location: C:\eclipse34M5\I20080108-1320\eclipse

- [✓] com.ibm.icu (3.6.1.v20070906)
- [✓] com.ibm.icu.source (3.6.1.v20070906)
- [✓] com.jcraft.jsch (0.1.31)
- [✓] javax.servlet (2.4.0.v200711021030)
- [✓] javax.servlet.jsp (2.0.0.v200711021030)
- [✓] javax.servlet.jsp.source (2.0.0.v200711021030)
- [✓] javax.servlet.source (2.4.0.v200711021030)
- [✓] org.apache.ant (1.7.0.v200711021030)
- [✓] org.apache.ant.source (1.7.0.v200711021030)
- [✓] org.apache.commons.el (1.0.0.v200711021015)
- [✓] org.apache.commons.el.source (1.0.0.v200711021015)
- [✓] org.apache.commons.logging (1.0.4.v200711021015)
- [✓] org.apache.commons.logging.source (1.0.4.v200711021015)
- [✓] org.apache.jasper (5.5.17.v200711021015)

251 of 251 selected.

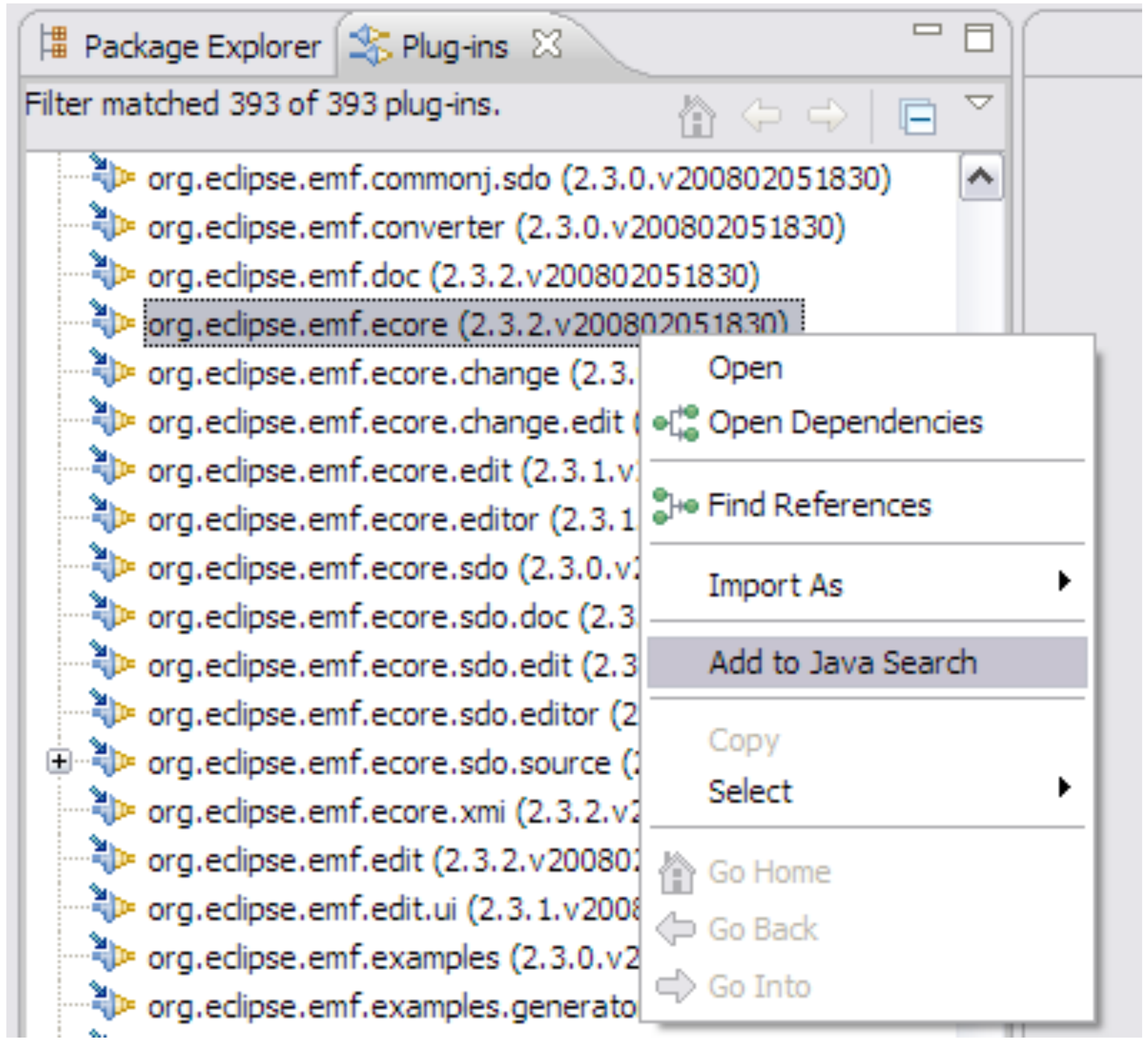
[ ] Group plug-ins by location

Pre-defined Targets

Target:

Plug-ins View

- A view into all the plug-ins you're working with
- Has the ability to add things to java search which can help you find classes via Ctrl+Shift +T

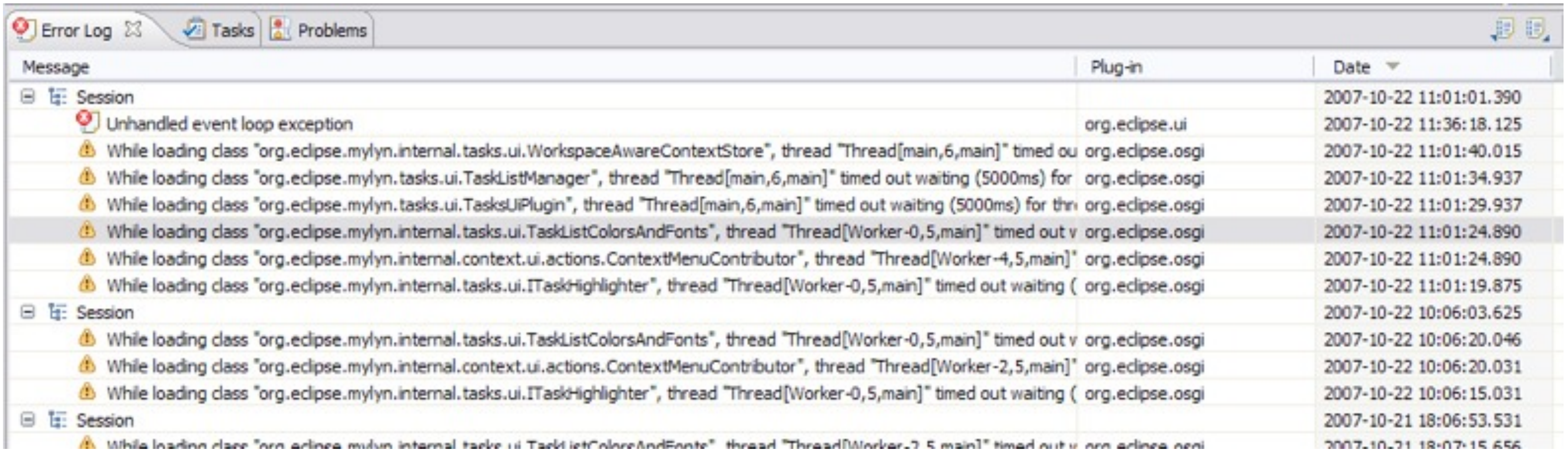


Filter matched 393 of 393 plug-ins.

```
org.eclipse.emf.commonj.sdo (2.3.0.v200802051830)
org.eclipse.emf.converter (2.3.0.v200802051830)
org.eclipse.emf.doc (2.3.2.v200802051830)
org.eclipse.emf.ecore (2.3.2.v200802051830)
org.eclipse.emf.ecore.change (2.3.
org.eclipse.emf.ecore.change.edit (
org.eclipse.emf.ecore.edit (2.3.1.v
org.eclipse.emf.ecore.editor (2.3.1
org.eclipse.emf.ecore.sdo (2.3.0.v
org.eclipse.emf.ecore.sdo.doc (2.3
org.eclipse.emf.ecore.sdo.edit (2.3
org.eclipse.emf.ecore.sdo.editor (2
org.eclipse.emf.ecore.sdo.source (
org.eclipse.emf.ecore.xmi (2.3.2.v
org.eclipse.emf.edit (2.3.2.v20080
org.eclipse.emf.edit.ui (2.3.1.v200
org.eclipse.emf.examples (2.3.0.v2
org.eclipse.emf.examples.generato
```

Error Log

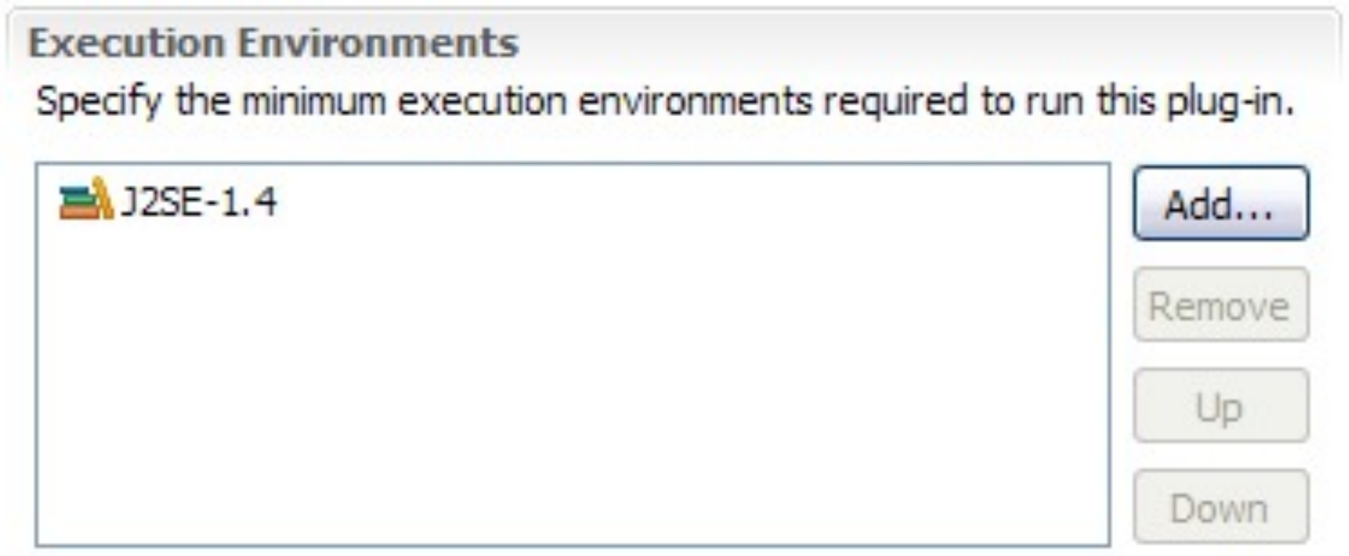
- More than meets the eye…
- Group log entries by
  - Session
  - Plug-in



Message	Plug-in	Date
Session		2007-10-22 11:01:01.390
Unhandled event loop exception	org.eclipse.ui	2007-10-22 11:36:18.125
While loading class "org.eclipse.mylyn.internal.tasks.ui.WorkspaceAwareContextStore", thread "Thread[main,6,main]" timed ou	org.eclipse.osgi	2007-10-22 11:01:40.015
While loading class "org.eclipse.mylyn.tasks.ui.TaskListManager", thread "Thread[main,6,main]" timed out waiting (5000ms) for	org.eclipse.osgi	2007-10-22 11:01:34.937
While loading class "org.eclipse.mylyn.tasks.ui.TasksUiPlugin", thread "Thread[main,6,main]" timed out waiting (5000ms) for thr	org.eclipse.osgi	2007-10-22 11:01:29.937
While loading class "org.eclipse.mylyn.internal.tasks.ui.TaskListColorsAndFonts", thread "Thread[Worker-0,5,main]" timed out v	org.eclipse.osgi	2007-10-22 11:01:24.890
While loading class "org.eclipse.mylyn.internal.context.ui.actions.ContextMenuContributor", thread "Thread[Worker-4,5,main]"	org.eclipse.osgi	2007-10-22 11:01:24.890
While loading class "org.eclipse.mylyn.internal.tasks.ui.ITaskHighlighter", thread "Thread[Worker-0,5,main]" timed out waiting (	org.eclipse.osgi	2007-10-22 11:01:19.875
Session		2007-10-22 10:06:03.625
While loading class "org.eclipse.mylyn.internal.tasks.ui.TaskListColorsAndFonts", thread "Thread[Worker-0,5,main]" timed out v	org.eclipse.osgi	2007-10-22 10:06:20.046
While loading class "org.eclipse.mylyn.internal.context.ui.actions.ContextMenuContributor", thread "Thread[Worker-2,5,main]"	org.eclipse.osgi	2007-10-22 10:06:20.031
While loading class "org.eclipse.mylyn.internal.tasks.ui.ITaskHighlighter", thread "Thread[Worker-0,5,main]" timed out waiting (	org.eclipse.osgi	2007-10-22 10:06:15.031
Session		2007-10-21 18:06:53.531
While loading class "org.eclipse.mylyn.internal.tasks.ui.TaskListColorsAndFonts", thread "Thread[Worker-2,5,main]" timed out v	org.eclipse.osgi	2007-10-21 18:07:15.656

Execution Environments

- Execution Environments are symbolic representations of JREs
- Bundle-RequiredExecutionEnvironment (BREE) manifest header
- **PDE Build uses BREE to determine compile settings**
- http://wiki.eclipse.org/Execution_Environments



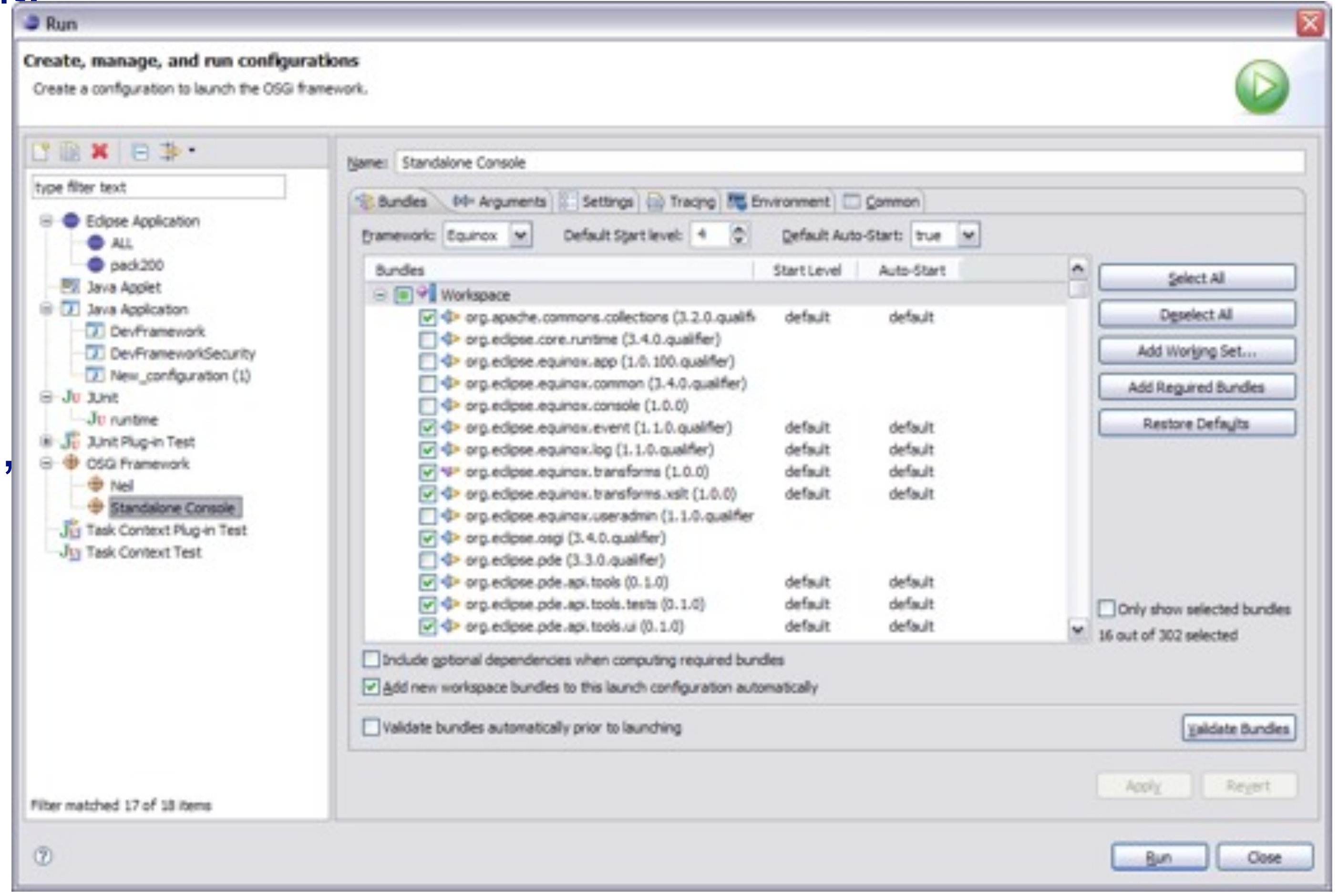
Execution Environments

Specify the minimum execution environments required to run this plug-in.

J2SE-1.4

OSGi Launch Configurations

- Provides a way to easily run and test your bundle in an OSGi environment.
- Extensible framework that allows other OSGi runtimes to provide implementations to let users test on runtimes other than Equinox
- Gives users more advanced control, including the option to specify start levels for individual bundles.



Name: Standalone Console

Framework: Equinox Default Start level: 4 Default Auto-Start: true

Bundles	Start Level	Auto-Start
Workspace		
✓ org.apache.commons.collections (3.2.0.qualifi	default	default
org.eclipse.core.runtime (3.4.0.qualifier)		
org.eclipse.equinox.app (1.0.100.qualifier)		
org.eclipse.equinox.common (3.4.0.qualifier)		
org.eclipse.equinox.console (1.0.0)		
✓ org.eclipse.equinox.event (1.1.0.qualifier)	default	default
✓ org.eclipse.equinox.log (1.1.0.qualifier)	default	default
✓ org.eclipse.equinox.transforms (1.0.0)	default	default
✓ org.eclipse.equinox.transforms.xslt (1.0.0)	default	default
org.eclipse.equinox.useradmin (1.1.0.qualifier		
✓ org.eclipse.osgi (3.4.0.qualifier)		
org.eclipse.pde (3.3.0.qualifier)		
✓ org.eclipse.pde.api.tools (0.1.0)	default	default
✓ org.eclipse.pde.api.tools.tests (0.1.0)	default	default
✓ org.eclipse.pde.api.tools.ui (0.1.0)	default	default

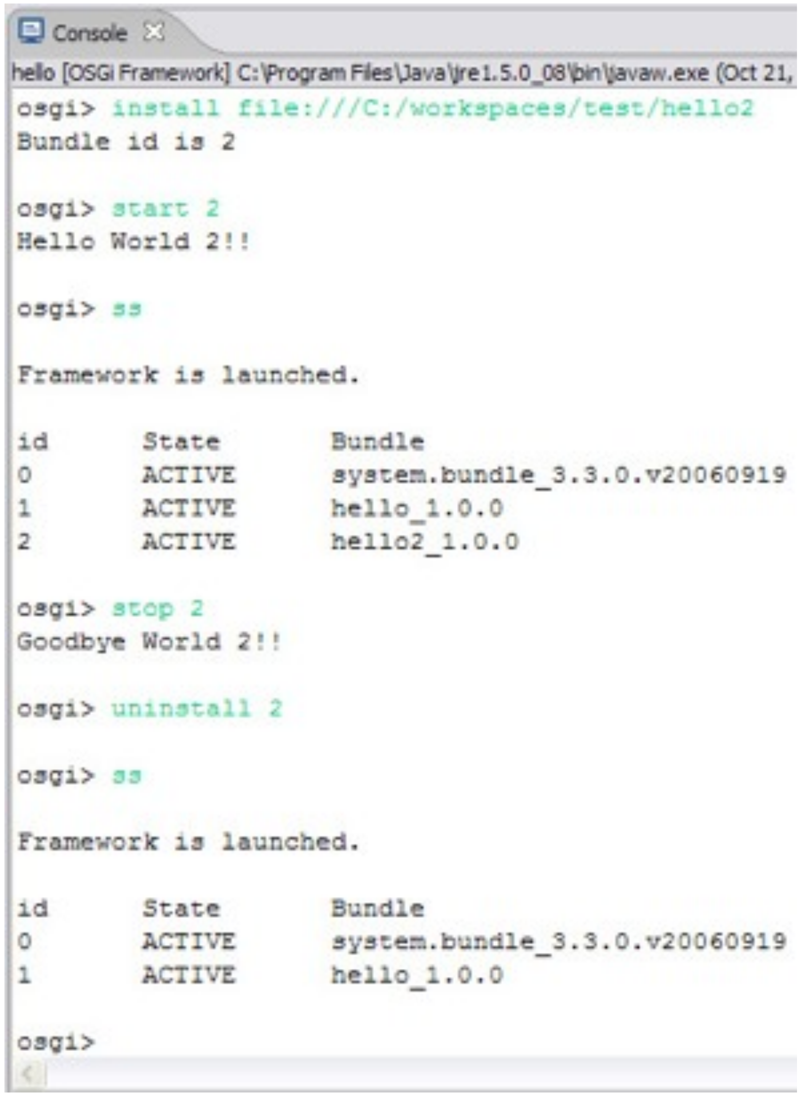
☐ Only show selected bundles
16 out of 302 selected

☐ Include optional dependencies when computing required bundles
☑ Add new workspace bundles to this launch configuration automatically
☐ Validate bundles automatically prior to launching

OSGi Console

- -console
- Integrate with the console that drives Eclipse
- Common commands
  - Status
  - Start/stop
  - Install/uninstall
  - diag
- Custom Commands
- http://www-128.ibm.com/developerworks/opensource/library/os-ecl-osgiconsole/

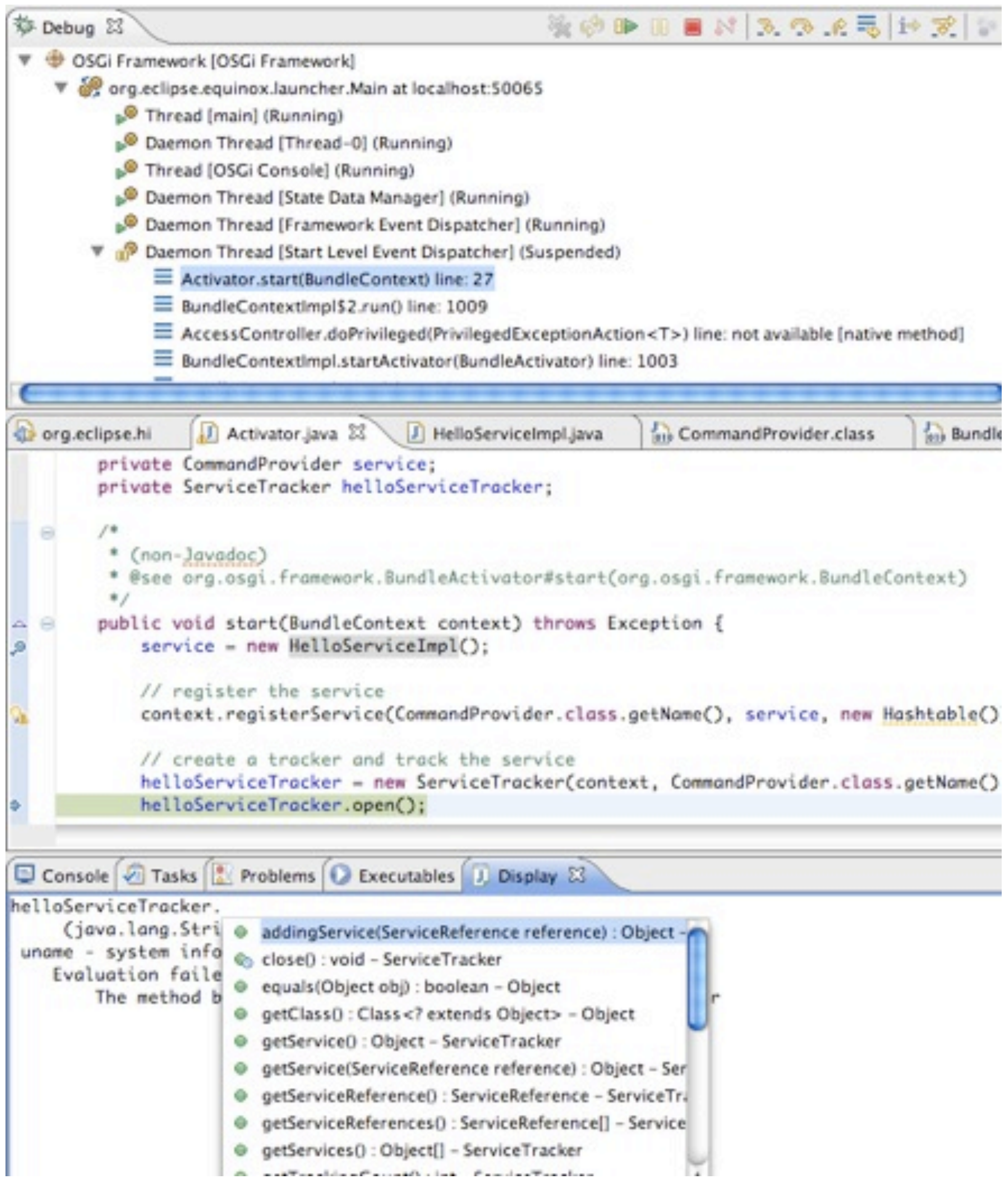
```
THE NECROPOLIS
MESHING PROGS/ARMOR.MDL...
YOU GOT THE SHELLS
YOU GOT THE GRENADE LAUNCHER
YOU RECEIVE 25 HEALTH
YOU GET 2 ROCKETS
YOU GOT THE ROCKETS
YOU GOT THE NAILGUN
YOU GOT THE NAILS
YOU GOT THE NAILS
YOU RECEIVE 15 HEALTH
YOU RECEIVE 15 HEALTH
YOU GOT THE ROCKETS
VERSION 1.09 SERVER (94778 CRC)

INTRODUCTION
]+MLOOK
MWHEELDOWN IS UNBOUND, HIT F4 TO SET.
WROTE QUAKE00.TGA
]
```



```
hello [OSGi Framework] C:\Program Files\Java\jre1.5.0_08\bin\javaw.exe (Oct 21,
osgi> install file:///C:/workspaces/test/hello2
Bundle id is 2

osgi> start 2
Hello World 2!!

osgi> ss

Framework is launched.

id      State       Bundle
0       ACTIVE      system.bundle_3.3.0.v20060919
1       ACTIVE      hello_1.0.0
2       ACTIVE      hello2_1.0.0

osgi> stop 2
Goodbye World 2!!

osgi> uninstall 2

osgi> ss

Framework is launched.

id      State       Bundle
0       ACTIVE      system.bundle_3.3.0.v20060919
1       ACTIVE      hello_1.0.0

osgi>
```

Display View

- The Display View allows you to execute/inspect code while you are stopped at a break point. This can be very helpful when trying to find specific objects or debug their contents.



```
OSGi Framework [OSGi Framework]
  org.eclipse.equinox.launcher.Main at localhost:50065
    Thread [main] (Running)
    Daemon Thread [Thread-0] (Running)
    Thread [OSGi Console] (Running)
    Daemon Thread [State Data Manager] (Running)
    Daemon Thread [Framework Event Dispatcher] (Running)
    Daemon Thread [Start Level Event Dispatcher] (Suspended)
      Activator.start(BundleContext) line: 27
      BundleContextImpl$2.run() line: 1009
      AccessController.doPrivileged(PrivilegedExceptionAction<T>) line: not available [native method]
      BundleContextImpl.startActivator(BundleActivator) line: 1003
```

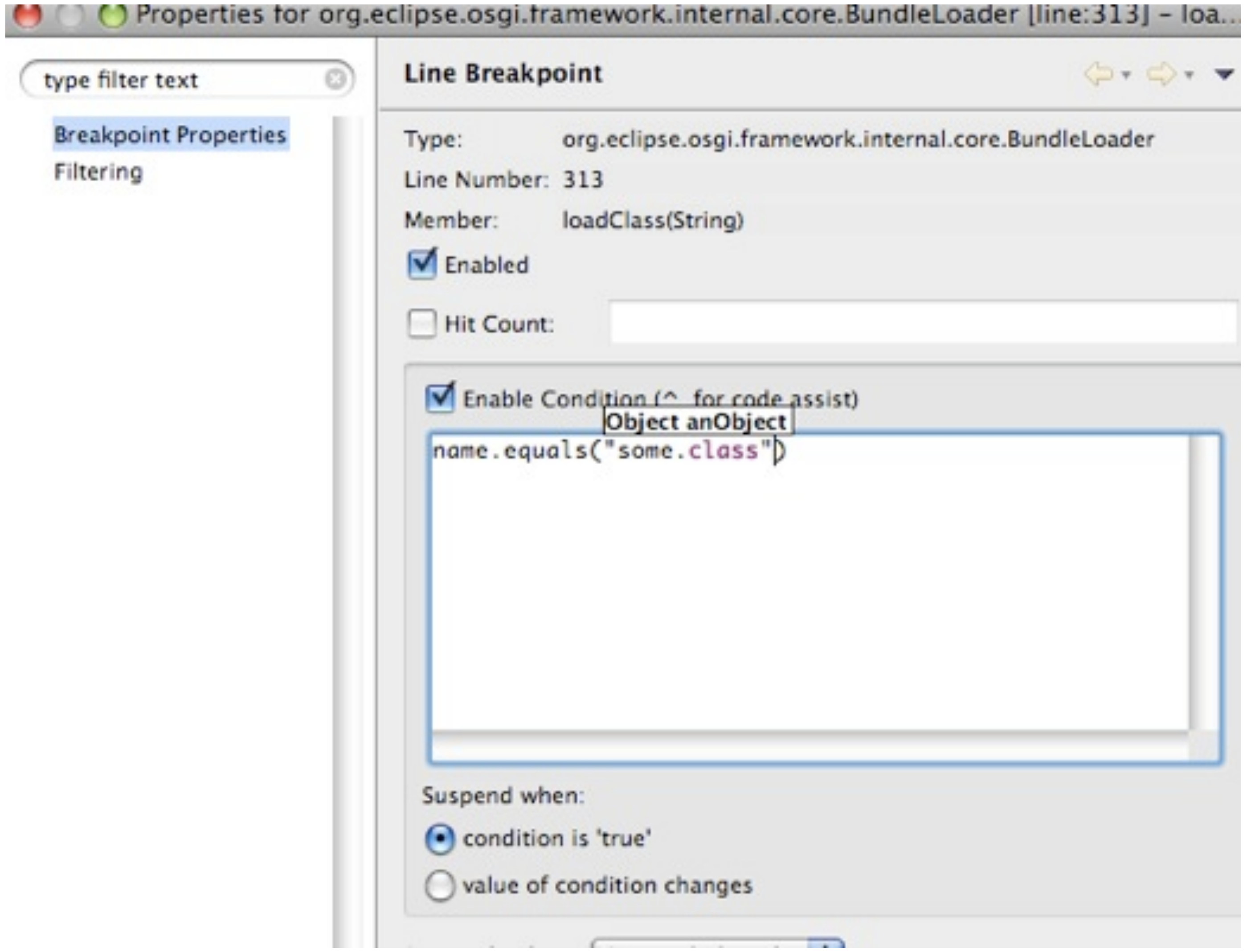
```java
    private CommandProvider service;
    private ServiceTracker helloServiceTracker;

    /*
     * (non-Javadoc)
     * @see org.osgi.framework.BundleActivator#start(org.osgi.framework.BundleContext)
     */
    public void start(BundleContext context) throws Exception {
        service = new HelloServiceImpl();

        // register the service
        context.registerService(CommandProvider.class.getName(), service, new Hashtable()

        // create a tracker and track the service
        helloServiceTracker = new ServiceTracker(context, CommandProvider.class.getName()
        helloServiceTracker.open();
```

```
helloServiceTracker.
    (java.lang.Stri
 uname - system info
   Evaluation faile
       The method b
```

```
addingService(ServiceReference reference) : Object -
close() : void - ServiceTracker
equals(Object obj) : boolean - Object
getClass() : Class<? extends Object> - Object
getService() : Object - ServiceTracker
getService(ServiceReference reference) : Object - Ser
getServiceReference() : ServiceReference - ServiceTr
getServiceReferences() : ServiceReference[] - Service
getServices() : Object[] - ServiceTracker
```

Conditional Breakpoints

- You can enable any breakpoint to stop only when it meets certain criteria. This helps if you are trying to iterate through a collection or if you are debugging a function which is called multiple times.


Properties for org.eclipse.osgi.framework.internal.core.BundleLoader [line:313] – loa...

type filter text

Breakpoint Properties
Filtering

Line Breakpoint

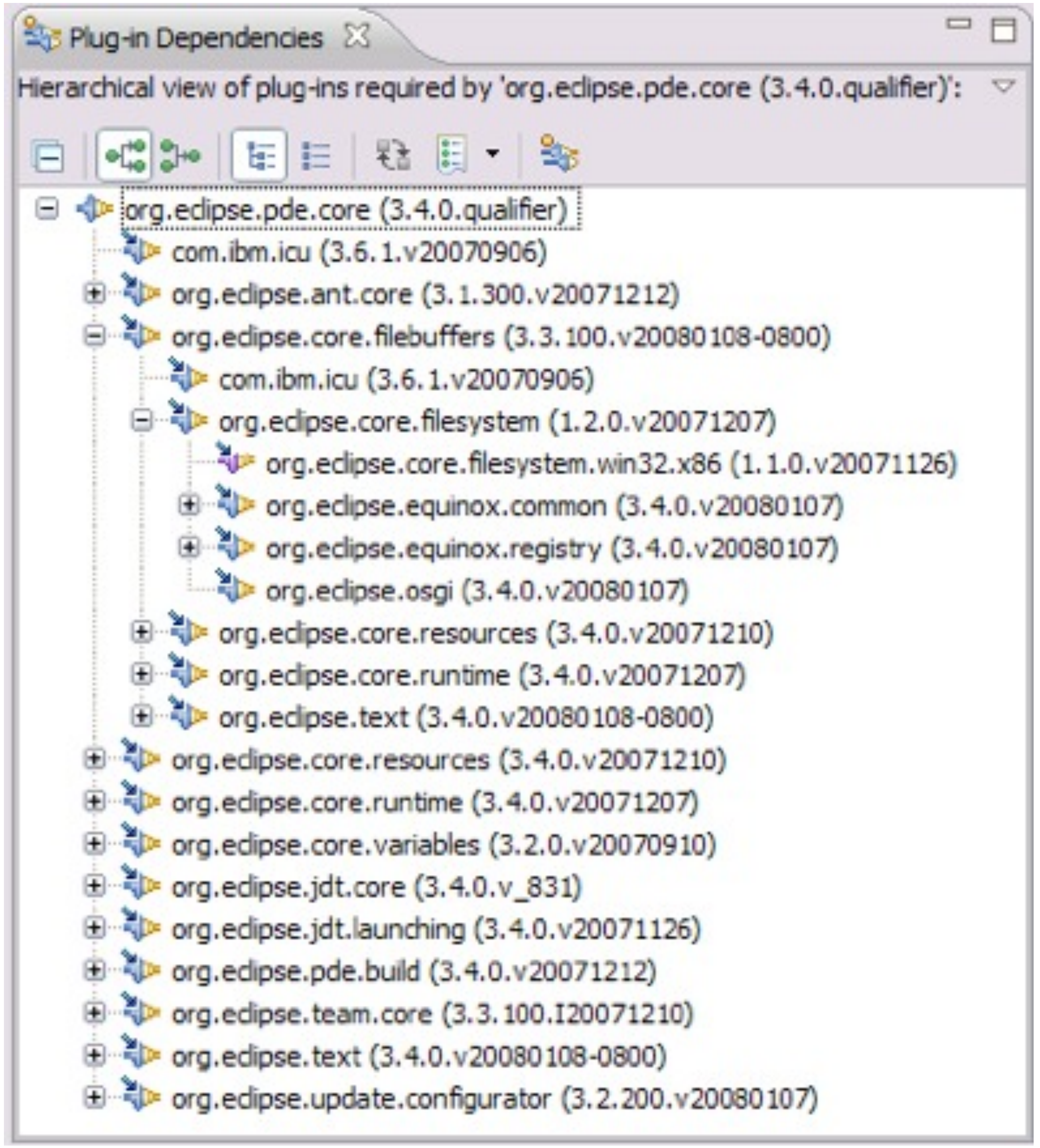
Type: org.eclipse.osgi.framework.internal.core.BundleLoader
Line Number: 313
Member: loadClass(String)
[✓] Enabled
[ ] Hit Count:
[✓] Enable Condition (^ for code assist)
Object anObject

```
name.equals("some.class")
```

Suspend when:
[● condition is 'true']
○ value of condition changes

Plug-in Dependencies View

- The **Plug-in Dependencies** view allows you to see all the dependencies for any plug-in project.
- You not only see what plug-ins a project depends on, but also what plug-ins depend on that project (callers and callees).
- It also can display the current state, including dependency wiring, of the plug-ins in the workspace and target platform. This will aid in finding resolution problems.

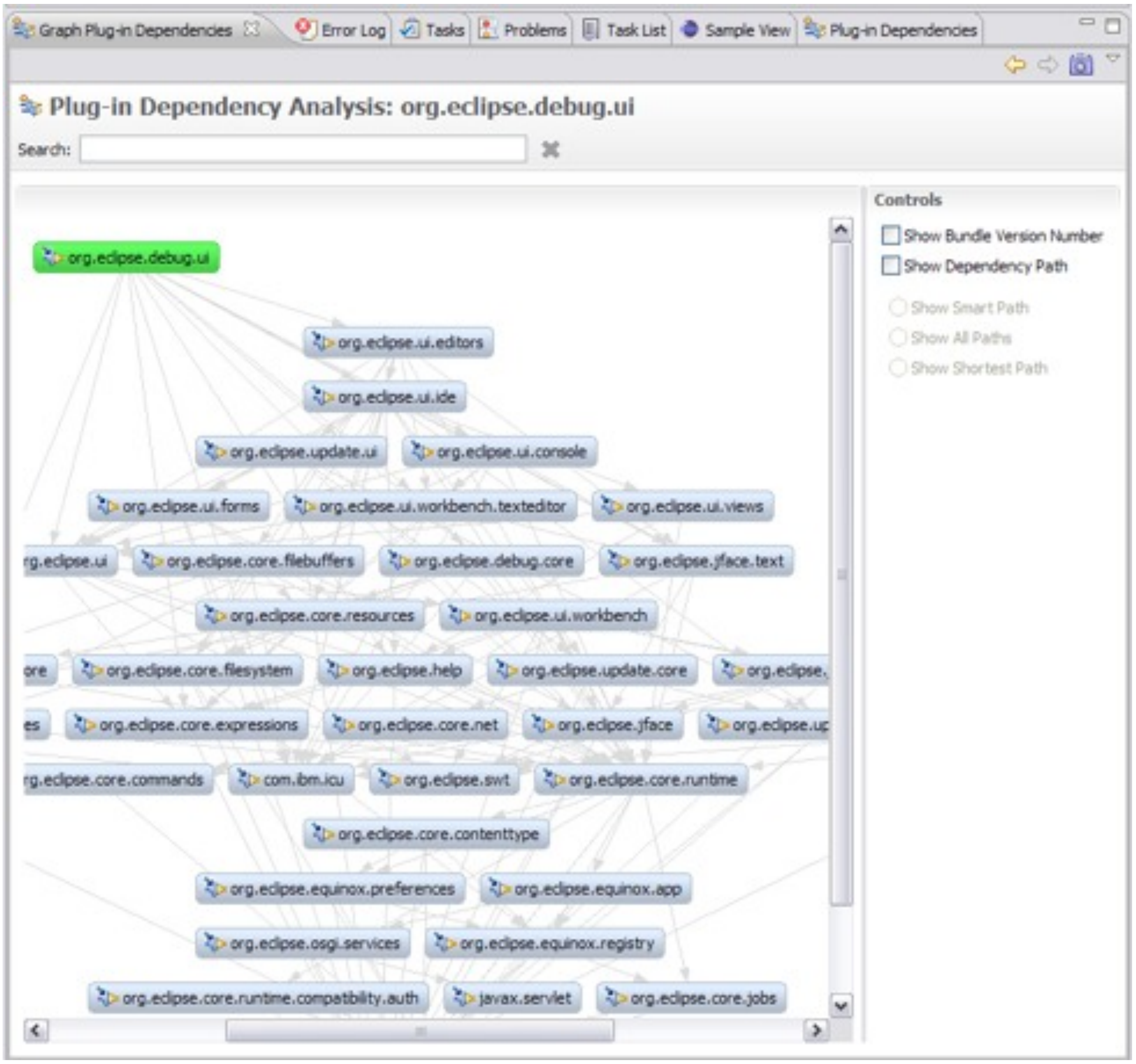


Hierarchical view of plug-ins required by 'org.eclipse.pde.core (3.4.0.qualifier)':

```
org.eclipse.pde.core (3.4.0.qualifier)
  com.ibm.icu (3.6.1.v20070906)
  org.eclipse.ant.core (3.1.300.v20071212)
  org.eclipse.core.filebuffers (3.3.100.v20080108-0800)
    com.ibm.icu (3.6.1.v20070906)
    org.eclipse.core.filesystem (1.2.0.v20071207)
      org.eclipse.core.filesystem.win32.x86 (1.1.0.v20071126)
      org.eclipse.equinox.common (3.4.0.v20080107)
      org.eclipse.equinox.registry (3.4.0.v20080107)
      org.eclipse.osgi (3.4.0.v20080107)
    org.eclipse.core.resources (3.4.0.v20071210)
    org.eclipse.core.runtime (3.4.0.v20071207)
    org.eclipse.text (3.4.0.v20080108-0800)
  org.eclipse.core.resources (3.4.0.v20071210)
  org.eclipse.core.runtime (3.4.0.v20071207)
  org.eclipse.core.variables (3.2.0.v20070910)
  org.eclipse.jdt.core (3.4.0.v_831)
  org.eclipse.jdt.launching (3.4.0.v20071126)
  org.eclipse.pde.build (3.4.0.v20071212)
  org.eclipse.team.core (3.3.100.I20071210)
  org.eclipse.text (3.4.0.v20080108-0800)
  org.eclipse.update.configurator (3.2.200.v20080107)
```

Graph Plug-in Dependencies View

- PDE Incubator Project
- Visualize your dependencies
- Pictures are worth a thousand words



Plug-in Dependency Analysis: org.eclipse.debug.ui

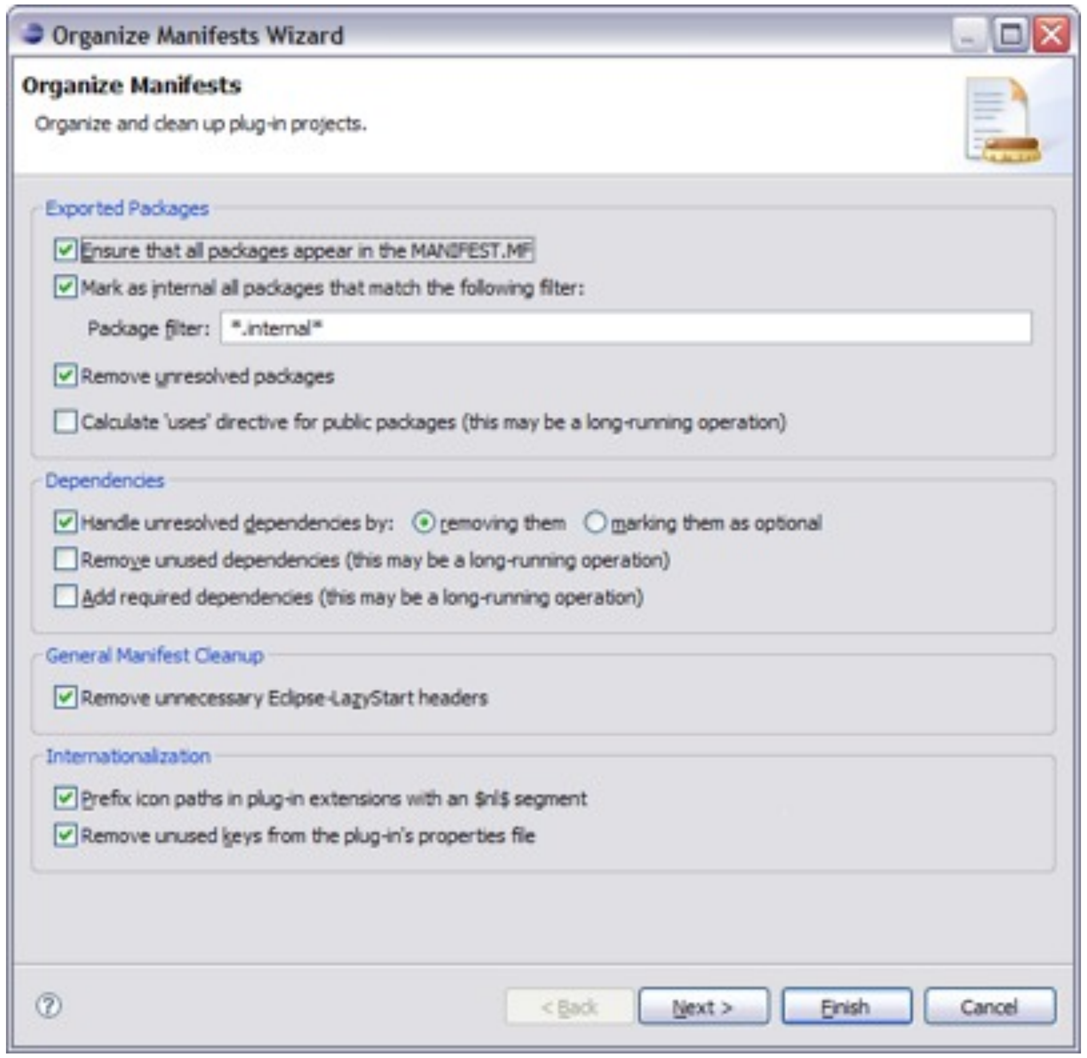
Controls

- ☐ Show Bundle Version Number
- ☐ Show Dependency Path
- ○ Show Smart Path
- ○ Show All Paths
- ○ Show Shortest Path

* *http://www.eclipse.org/pde/incubator/dependency-visualization/*

Organize Manifests Wizard

- Helps you modify and format your MANIFEST.MFs
- Accessible by right clicking on a MANIFEST.MF and selecting **PDE Tools > Organize Manifest**
- Helps you:
  - Export all packages in a project
  - Remove unresolved packages
  - Mark exported packages as internal
  - Modify unresolved dependencies (removing or making them optional)
  - Remove unused dependencies
  - Calculate dependencies (using Dependency Management)
  - Prefix icon paths with nl
  - Remove unused NLS keys
  - Calculate ‘uses’ directives



Organize Manifests Wizard

Organize Manifests

Organize and clean up plug-in projects.

Exported Packages
- [✓] Ensure that all packages appear in the MANIFEST.MF
- [✓] Mark as internal all packages that match the following filter:
  - Package filter: *.internal*
- [✓] Remove unresolved packages
- [ ] Calculate 'uses' directive for public packages (this may be a long-running operation)

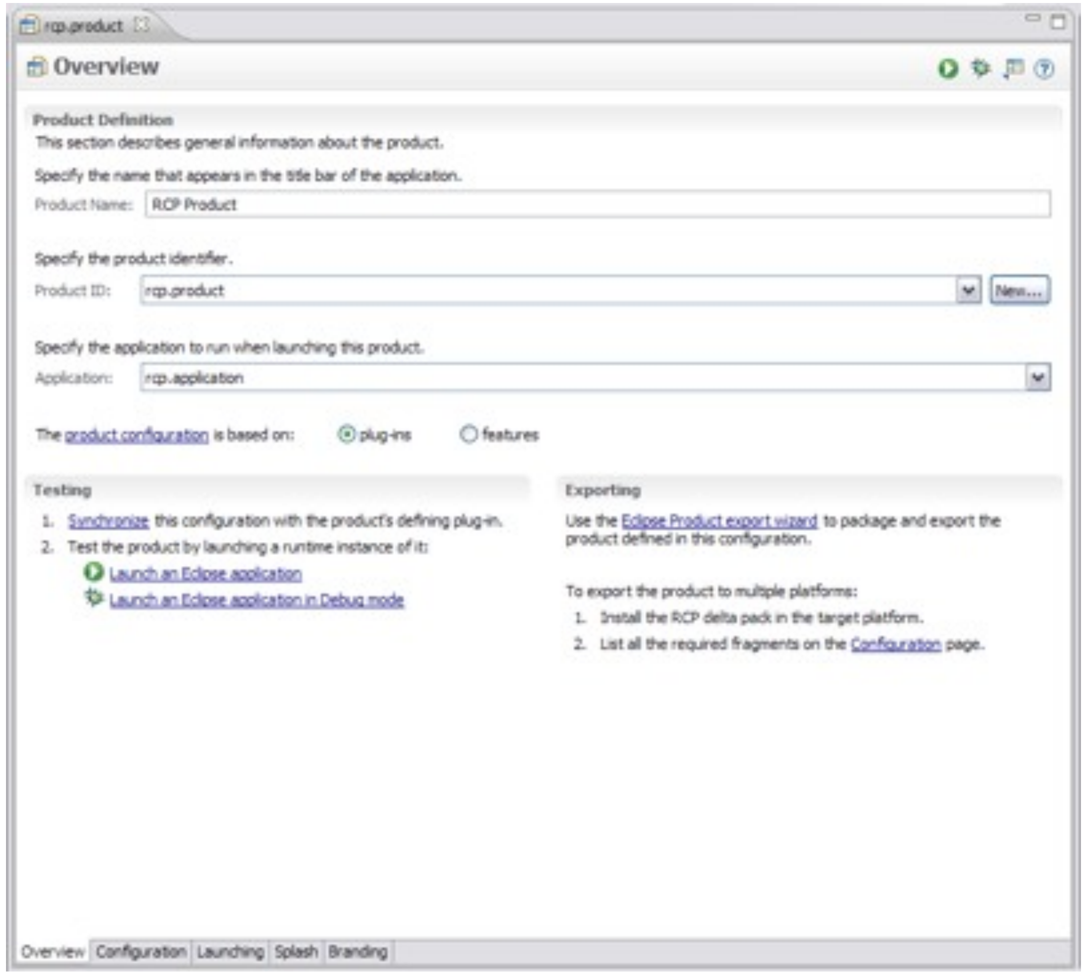
Dependencies
- [✓] Handle unresolved dependencies by: [✓ removing them] marking them as optional
- [ ] Remove unused dependencies (this may be a long-running operation)
- [ ] Add required dependencies (this may be a long-running operation)

General Manifest Cleanup
- [✓] Remove unnecessary Eclipse-LazyStart headers

Internationalization
- [✓] Prefix icon paths in plug-in extensions with an nl segment
- [✓] Remove unused keys from the plug-in's properties file

Product Editor

- A product definition helps you to easily customize, test and export an RCP/Eclipse application
- Customizations include:
  - Modifying which plug-ins are included
  - Create a splash screen
  - Bundling a JVM
  - Name for the launcher executable
  - Specify program and launching arguments
  - Define a welcome page and About dialog



Product Definition

This section describes general information about the product.

Specify the name that appears in the title bar of the application.

Product Name: RCP Product

Specify the product identifier.

Product ID: rcp.product

New...

Specify the application to run when launching this product.

Application: rcp.application

The product configuration is based on: [✓ plug-ins] features

Testing

1. Synchronize this configuration with the product's defining plug-in.
2. Test the product by launching a runtime instance of it:
   - Launch an Eclipse application
   - Launch an Eclipse application in Debug mode

Exporting

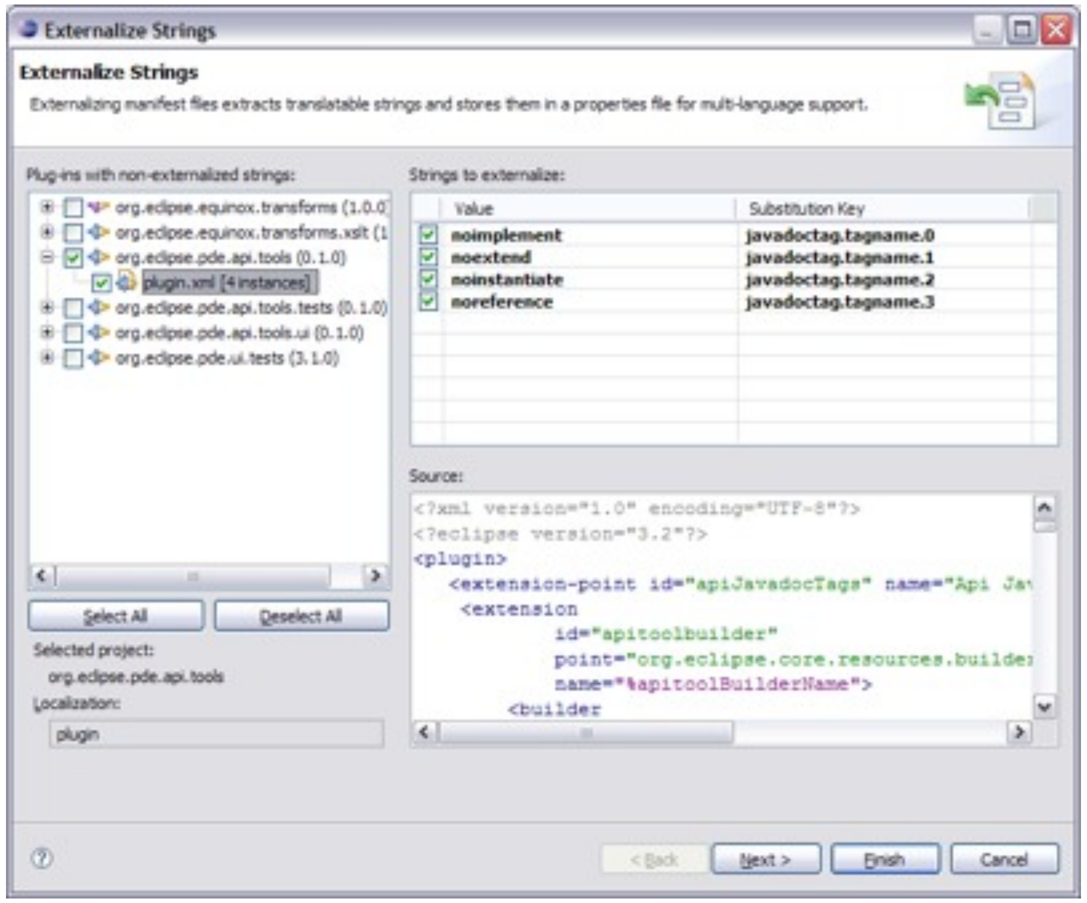
Use the Eclipse Product export wizard to package and export the product defined in this configuration.

To export the product to multiple platforms:

1. Install the RCP delta pack in the target platform.
2. List all the required fragments on the Configuration page.

Externalization Strings Wizard

- PDE’s Externalize Strings wizard allows you to quickly locate and painlessly externalize values in a plug-in’s MANIFEST.MF and plugin.xml.
- Accessible by right clicking on a MANIFEST.MF or plugin.xml and selecting **PDE Tools > Externalize Strings…**
- Externalized values are put in a file specified by the Bundle-Localization header. The default value for this file is “plugin.properties



Externalize Strings

Externalize Strings

Externalizing manifest files extracts translatable strings and stores them in a properties file for multi-language support.

Plug-ins with non-externalized strings:

- [ ] org.eclipse.equinox.transforms (1.0.0
- [ ] org.eclipse.equinox.transforms.xslt (1
- [✓] org.eclipse.pde.api.tools (0.1.0)
  - [✓] plugin.xml [4 instances]
- [ ] org.eclipse.pde.api.tools.tests (0.1.0)
- [ ] org.eclipse.pde.api.tools.ui (0.1.0)
- [ ] org.eclipse.pde.ui.tests (3.1.0)

Selected project: org.eclipse.pde.api.tools

Localization: plugin

Strings to externalize:

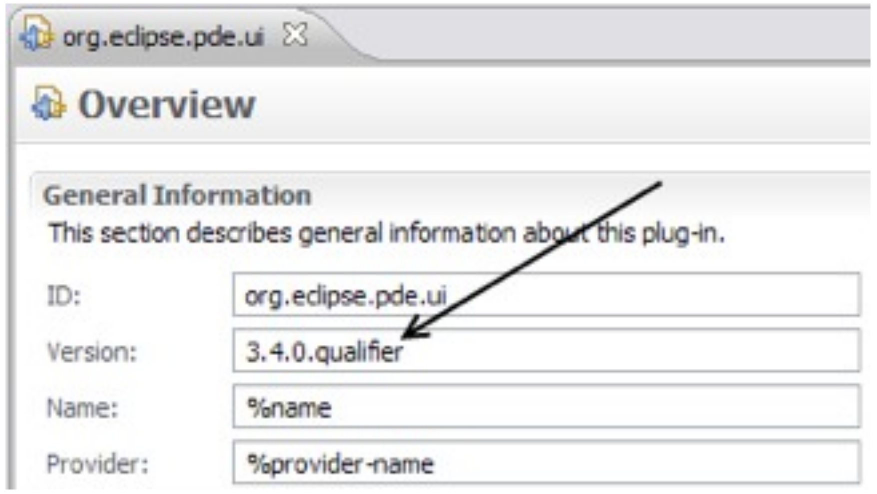
	Value	Substitution Key
✓	**noimplement**	**javadoctag.tagname.0**
✓	**noextend**	**javadoctag.tagname.1**
✓	**noinstantiate**	**javadoctag.tagname.2**
✓	**noreference**	**javadoctag.tagname.3**

Source:

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.2"?>
<plugin>
   <extension-point id="apiJavadocTags" name="Api Ja
    <extension
            id="apitoolbuilder"
            point="org.eclipse.core.resources.builde
            name="%apitoolBuilderName">
        <builder
```

.qualifier is awesome

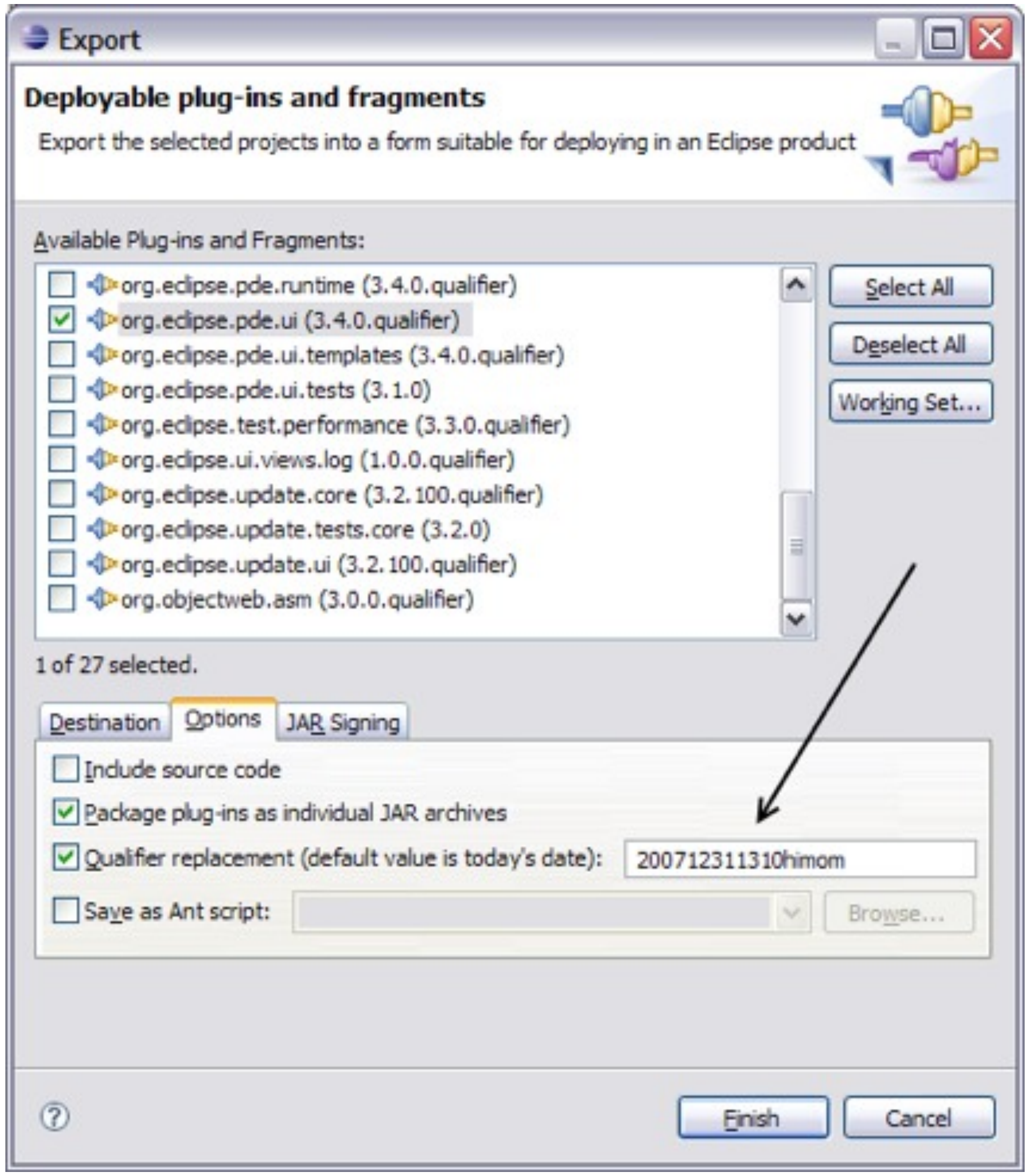
- The ".qualifier" marker allows you to easily substitute a value for the micro segment of a plug-in or feature's version.
- The date is the default value, but you substitute any value when exporting your project using the PDE export wizards.



General Information

This section describes general information about this plug-in.

ID:	org.eclipse.pde.ui
Version:	3.4.0.qualifier
Name:	%name
Provider:	%provider-name



Deployable plug-ins and fragments

Export the selected projects into a form suitable for deploying in an Eclipse product

Available Plug-ins and Fragments:

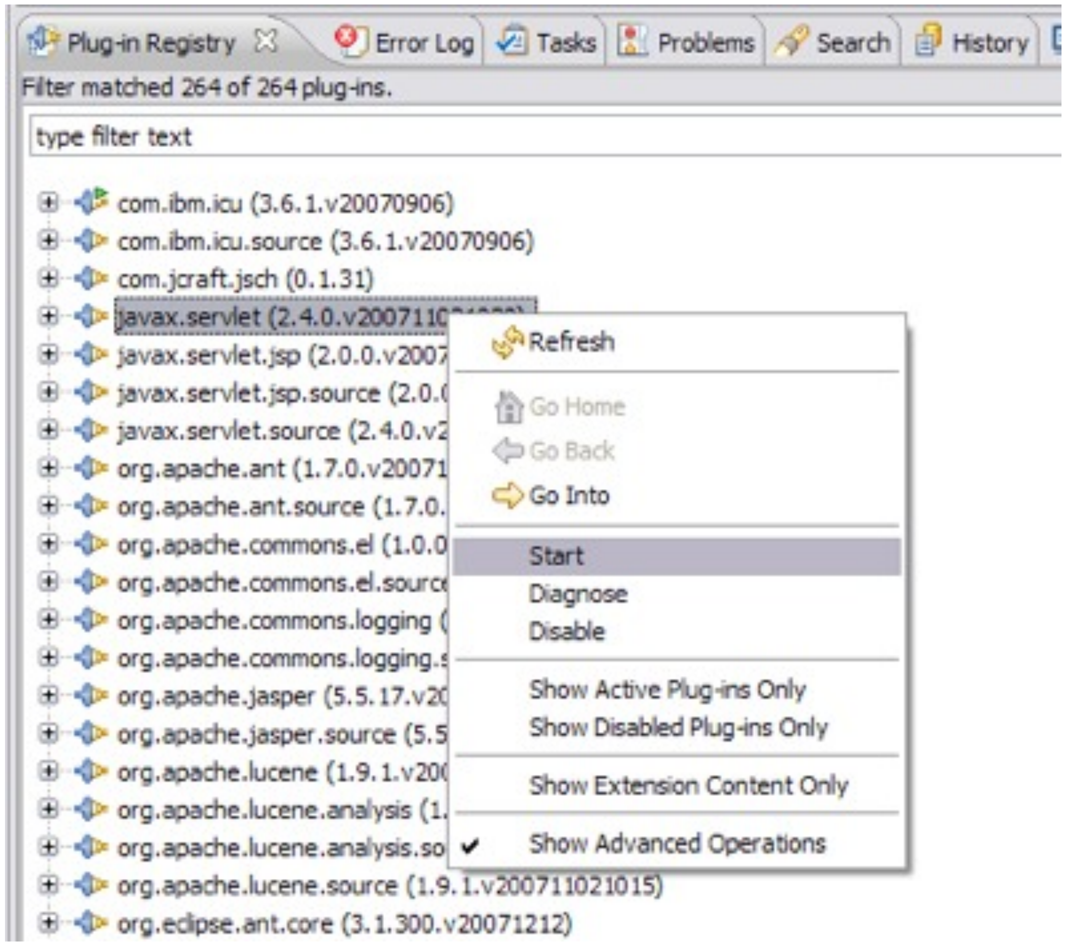
- [ ] org.eclipse.pde.runtime (3.4.0.qualifier)
- [x] org.eclipse.pde.ui (3.4.0.qualifier)
- [ ] org.eclipse.pde.ui.templates (3.4.0.qualifier)
- [ ] org.eclipse.pde.ui.tests (3.1.0)
- [ ] org.eclipse.test.performance (3.3.0.qualifier)
- [ ] org.eclipse.ui.views.log (1.0.0.qualifier)
- [ ] org.eclipse.update.core (3.2.100.qualifier)
- [ ] org.eclipse.update.tests.core (3.2.0)
- [ ] org.eclipse.update.ui (3.2.100.qualifier)
- [ ] org.objectweb.asm (3.0.0.qualifier)

1 of 27 selected.

- [ ] Include source code
- [x] Package plug-ins as individual JAR archives
- [x] Qualifier replacement (default value is today's date): 200712311310himom
- [ ] Save as Ant script:

Plug-in Registry View

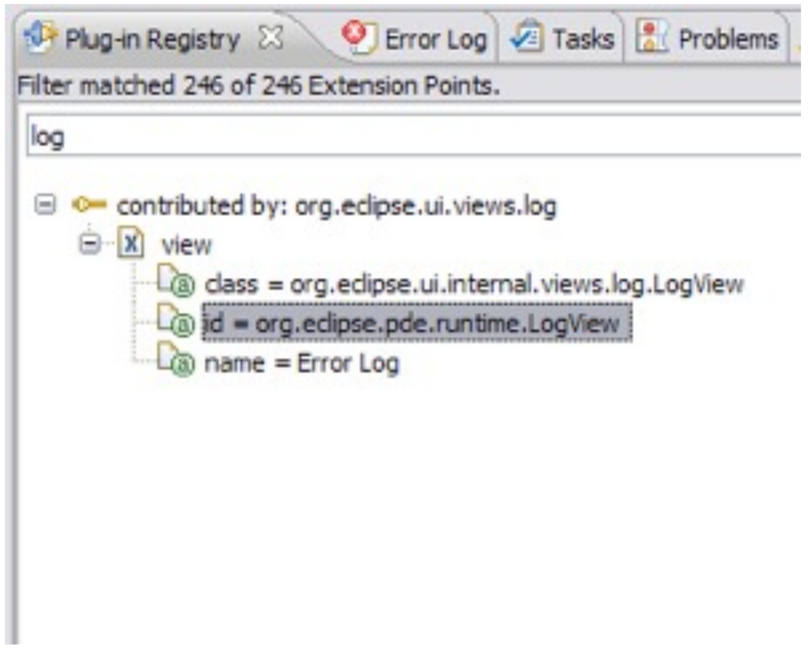
- The **Plug-in Registry** view is your eye into the runtime
- Show Advanced Operations
  - start/stop bundles
- Show Extension Content Only
  - quickly browse extensions


Filter matched 264 of 264 plug-ins.

type filter text

com.ibm.icu (3.6.1.v20070906)
com.ibm.icu.source (3.6.1.v20070906)
com.jcraft.jsch (0.1.31)
javax.servlet (2.4.0.v2007110
javax.servlet.jsp (2.0.0.v2007
javax.servlet.jsp.source (2.0.(
javax.servlet.source (2.4.0.v2
org.apache.ant (1.7.0.v20071
org.apache.ant.source (1.7.0.
org.apache.commons.el (1.0.0
org.apache.commons.el.source
org.apache.commons.logging (
org.apache.commons.logging.s
org.apache.jasper (5.5.17.v20
org.apache.jasper.source (5.5
org.apache.lucene (1.9.1.v200
org.apache.lucene.analysis (1.
org.apache.lucene.analysis.so
org.apache.lucene.source (1.9.1.v200711021015)
org.eclipse.ant.core (3.1.300.v20071212)

Plug-in Registry
Error Log
Tasks
Problems
Search
History
Refresh
Go Home
Go Back
Go Into
Start
Diagnose
Disable
Show Active Plug-ins Only
Show Disabled Plug-ins Only
Show Extension Content Only
✓ Show Advanced Operations

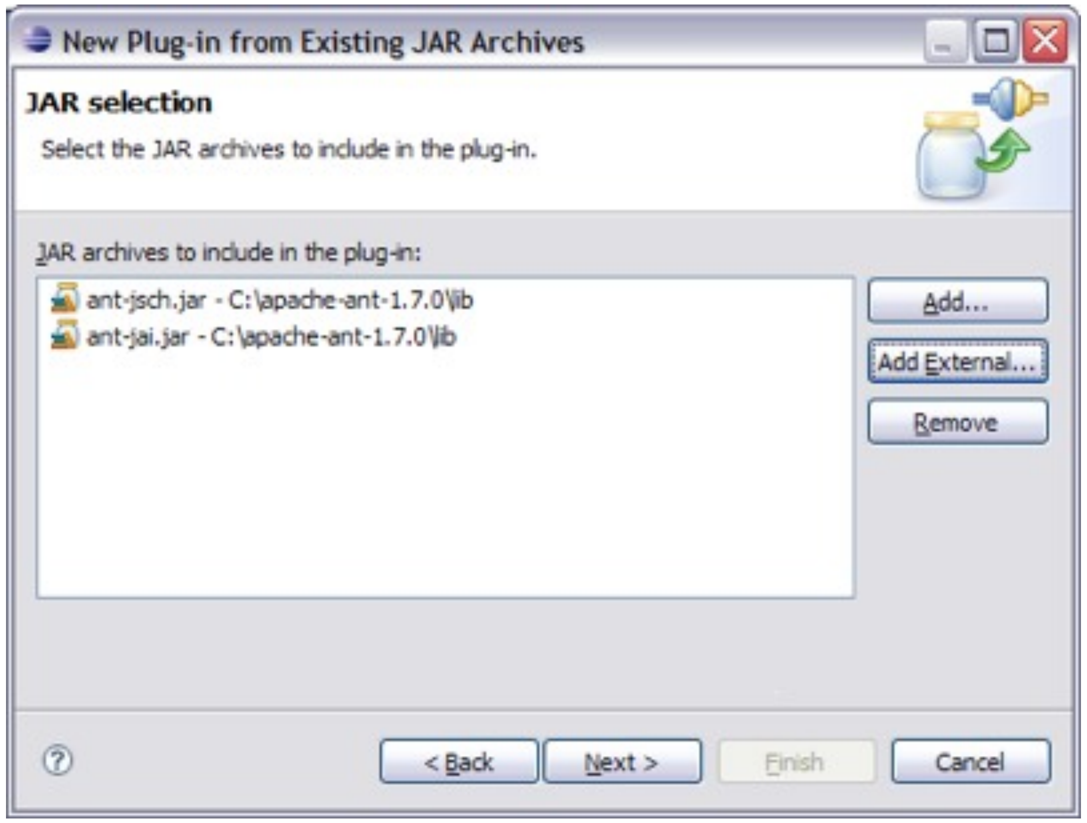

Filter matched 246 of 246 Extension Points.

log

contributed by: org.eclipse.ui.views.log
view
class = org.eclipse.ui.internal.views.log.LogView
id = org.eclipse.pde.runtime.LogView
name = Error Log

Plug-in Project from existing jars

- The **Plug-in Project from Existing Jars** wizard enables you to quickly convert jar files to plug-ins.
- Helpful when an application is being converted to OSGi and it depends on certain library jars
- Can be very useful for utility jars, as they can be shared across multiple plug-ins instead of requiring the jars be included in each plug-in
- Embedded JARs are evil


New Plug-in from Existing JAR Archives

JAR selection

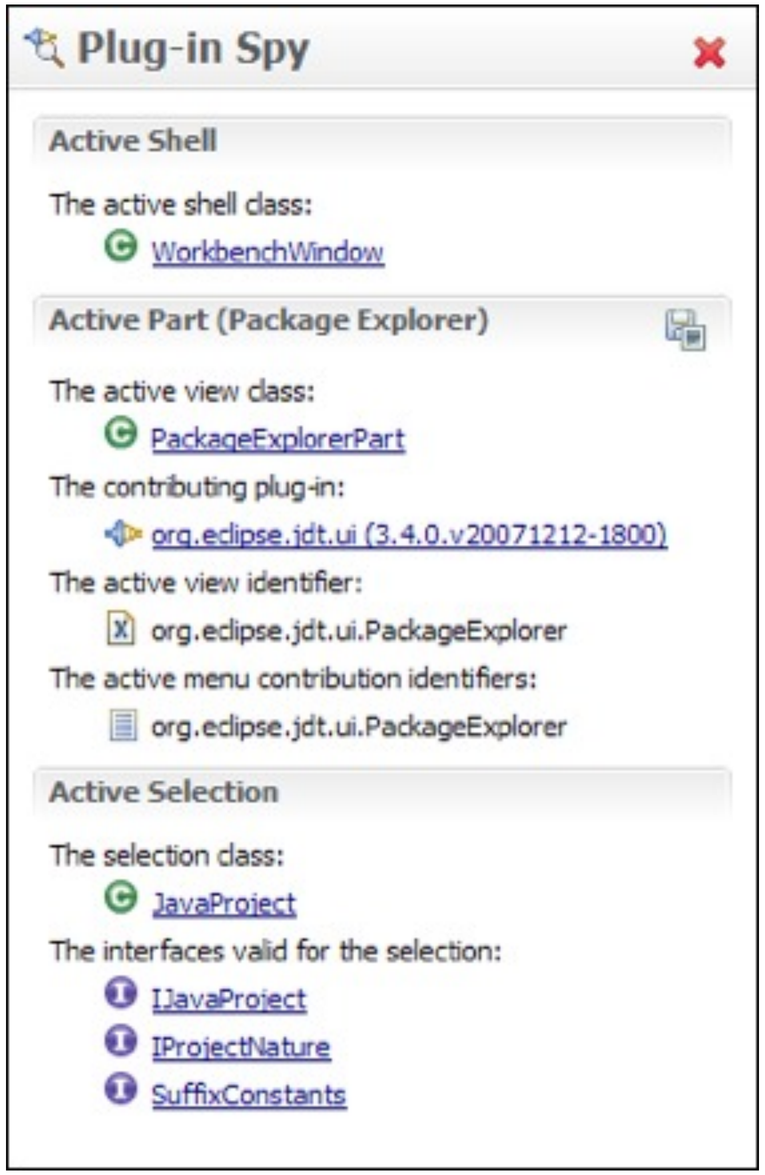
Select the JAR archives to include in the plug-in.

JAR archives to include in the plug-in:

ant-jsch.jar - C:\apache-ant-1.7.0\lib
ant-jai.jar - C:\apache-ant-1.7.0\lib

Plug-in Spy (3.4M3)

- ALT+SHIFT+F1
- Allows you to introspect what you're looking at…
- Hyperlinking
- Shows contributing plug-ins



Plug-in Spy

Active Shell

The active shell class:
WorkbenchWindow

Active Part (Package Explorer)

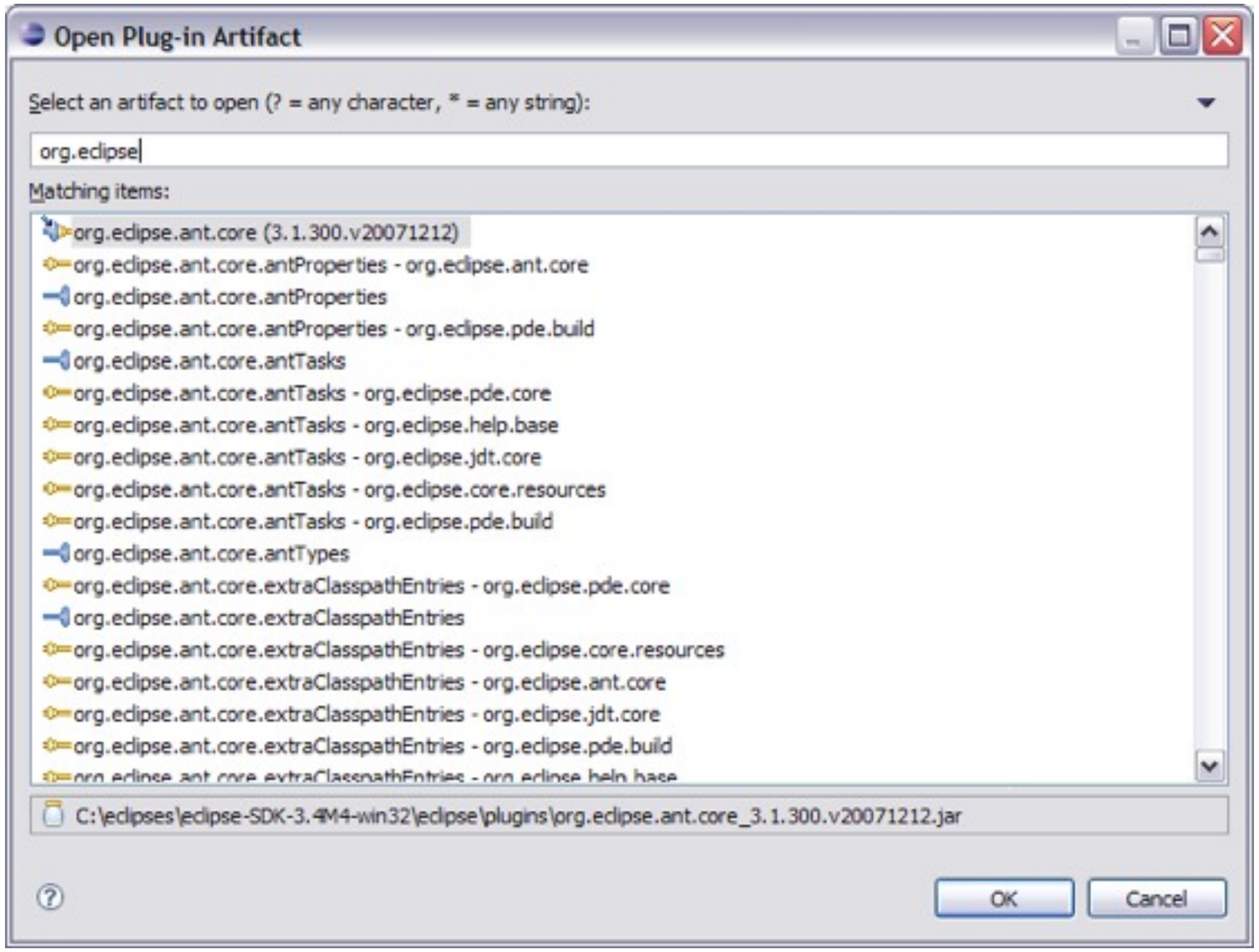
The active view class:
PackageExplorerPart
The contributing plug-in:
org.eclipse.jdt.ui (3.4.0.v20071212-1800)
The active view identifier:
org.eclipse.jdt.ui.PackageExplorer
The active menu contribution identifiers:
org.eclipse.jdt.ui.PackageExplorer

Active Selection

The selection class:
JavaProject
The interfaces valid for the selection:
IJavaProject
IProjectNature
SuffixConstants

Open Plug-in Artifact (3.4M4)

- Ctrl+Shift+A
- Quickly browse plug-ins, extensions and extension points



Select an artifact to open (? = any character, * = any string):

org.eclipse

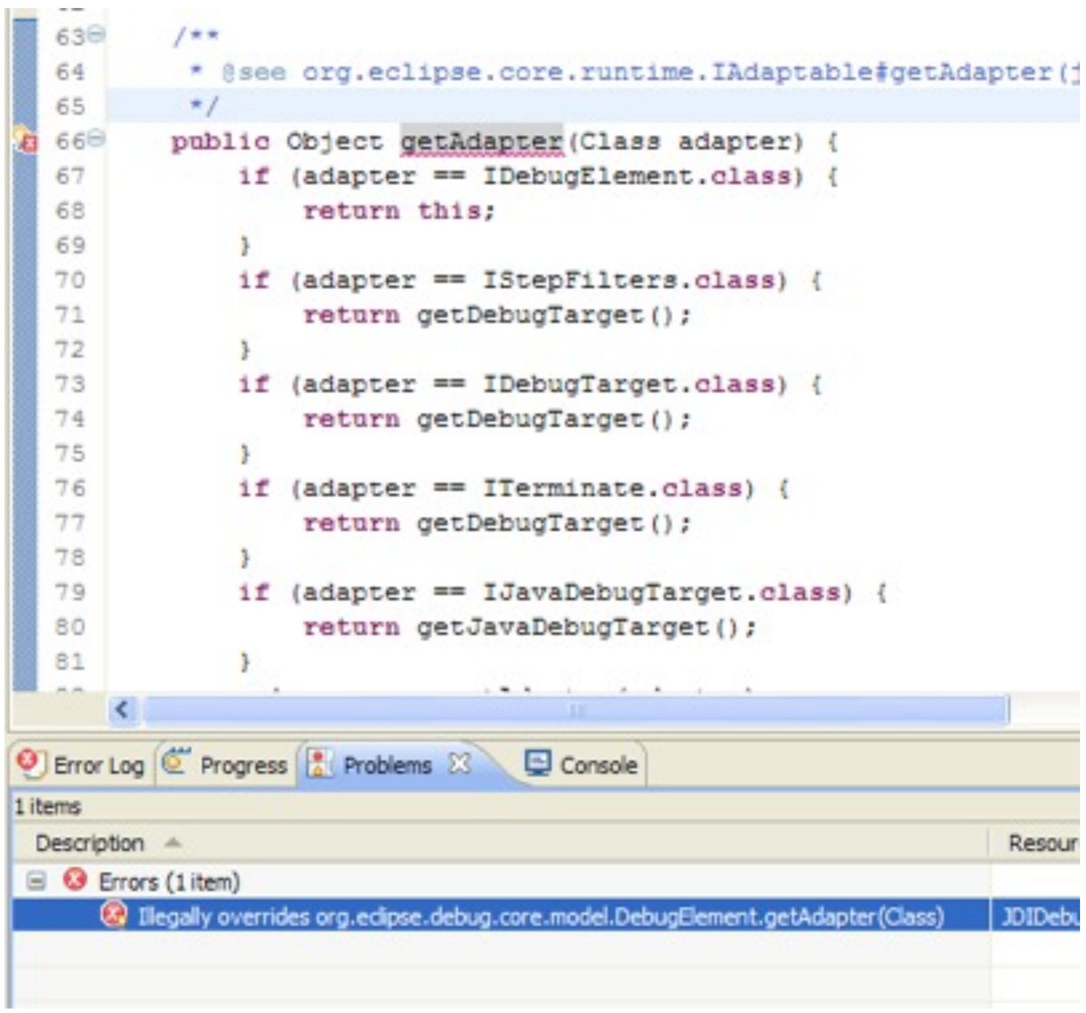
Matching items:

```
org.eclipse.ant.core (3.1.300.v20071212)
org.eclipse.ant.core.antProperties - org.eclipse.ant.core
org.eclipse.ant.core.antProperties
org.eclipse.ant.core.antProperties - org.eclipse.pde.build
org.eclipse.ant.core.antTasks
org.eclipse.ant.core.antTasks - org.eclipse.pde.core
org.eclipse.ant.core.antTasks - org.eclipse.help.base
org.eclipse.ant.core.antTasks - org.eclipse.jdt.core
org.eclipse.ant.core.antTasks - org.eclipse.core.resources
org.eclipse.ant.core.antTasks - org.eclipse.pde.build
org.eclipse.ant.core.antTypes
org.eclipse.ant.core.extraClasspathEntries - org.eclipse.pde.core
org.eclipse.ant.core.extraClasspathEntries
org.eclipse.ant.core.extraClasspathEntries - org.eclipse.core.resources
org.eclipse.ant.core.extraClasspathEntries - org.eclipse.ant.core
org.eclipse.ant.core.extraClasspathEntries - org.eclipse.jdt.core
org.eclipse.ant.core.extraClasspathEntries - org.eclipse.pde.build
org.eclipse.ant.core.extraClasspathEntries - org.eclipse.help.base
```

C:\eclipses\eclipse-SDK-3.4M4-win32\eclipse\plugins\org.eclipse.ant.core_3.1.300.v20071212.jar

API Tooling (3.4M6)

- API tooling will assist developers in API maintenance by reporting…
  - API defects such as binary incompatibilities
  - incorrect plug-in version numbers
  - missing or incorrect @since tags
  - usage of non-API code between plug-ins



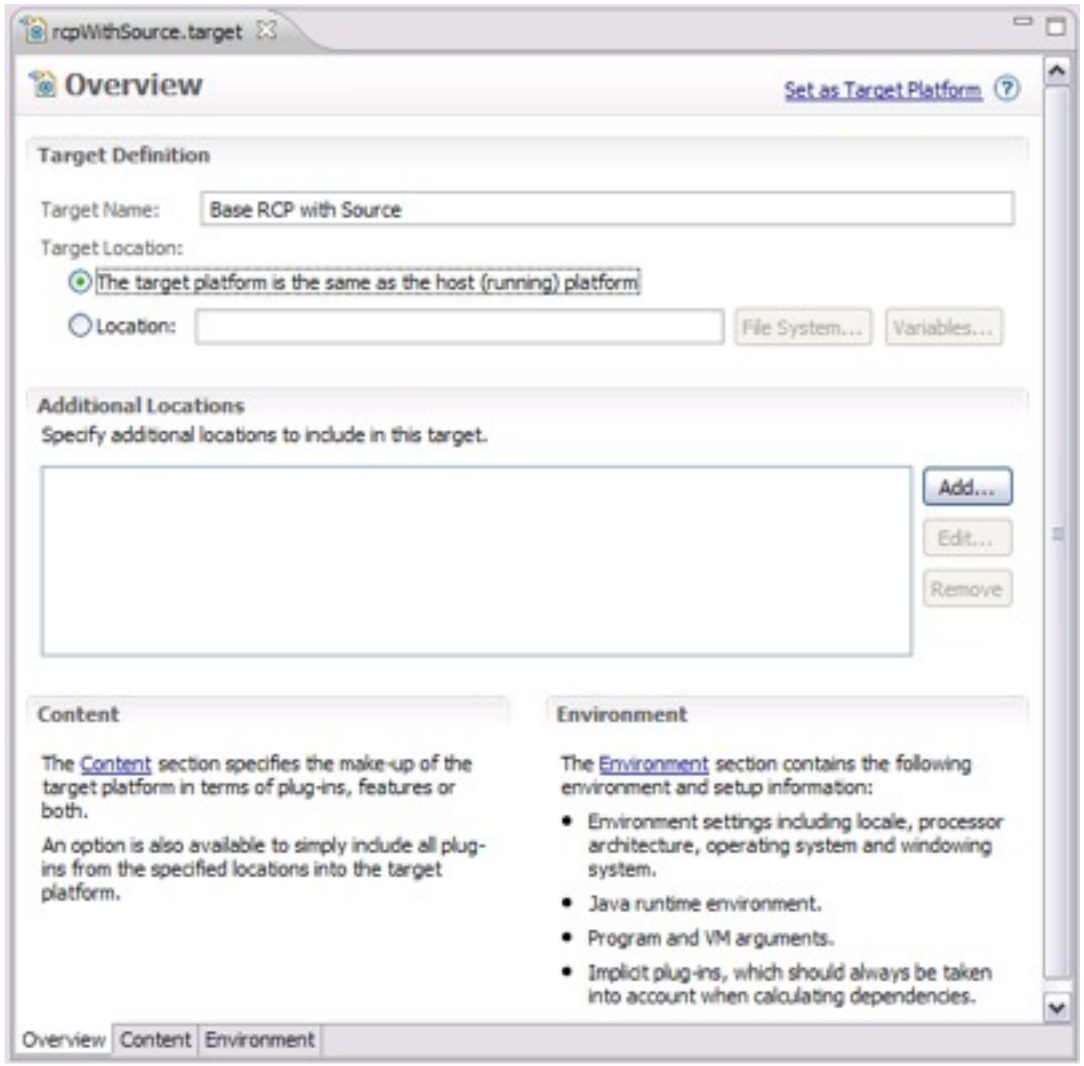
```
    /**
     * @see org.eclipse.core.runtime.IAdaptable#getAdapter(j
     */
    public Object getAdapter(Class adapter) {
        if (adapter == IDebugElement.class) {
            return this;
        }
        if (adapter == IStepFilters.class) {
            return getDebugTarget();
        }
        if (adapter == IDebugTarget.class) {
            return getDebugTarget();
        }
        if (adapter == ITerminate.class) {
            return getDebugTarget();
        }
        if (adapter == IJavaDebugTarget.class) {
            return getJavaDebugTarget();
        }
```

1 items

Description	Resour
Errors (1 item)	
Illegally overrides org.eclipse.debug.core.model.DebugElement.getAdapter(Class)	JDIDebu

Target Editor

- A target definition if a file that helps to configure your PDE development environment.
- They can be created in the workspace or loaded from plug-ins who define them in your platform.



rcpWithSource.target

Overview

Set as Target Platform

Target Definition

Target Name: Base RCP with Source

Target Location:

[✓ The target platform is the same as the host (running) platform]

Location:

File System... Variables...

Additional Locations

Specify additional locations to include in this target.

Add... Edit... Remove

Content

The Content section specifies the make-up of the target platform in terms of plug-ins, features or both.

An option is also available to simply include all plug-ins from the specified locations into the target platform.

Environment

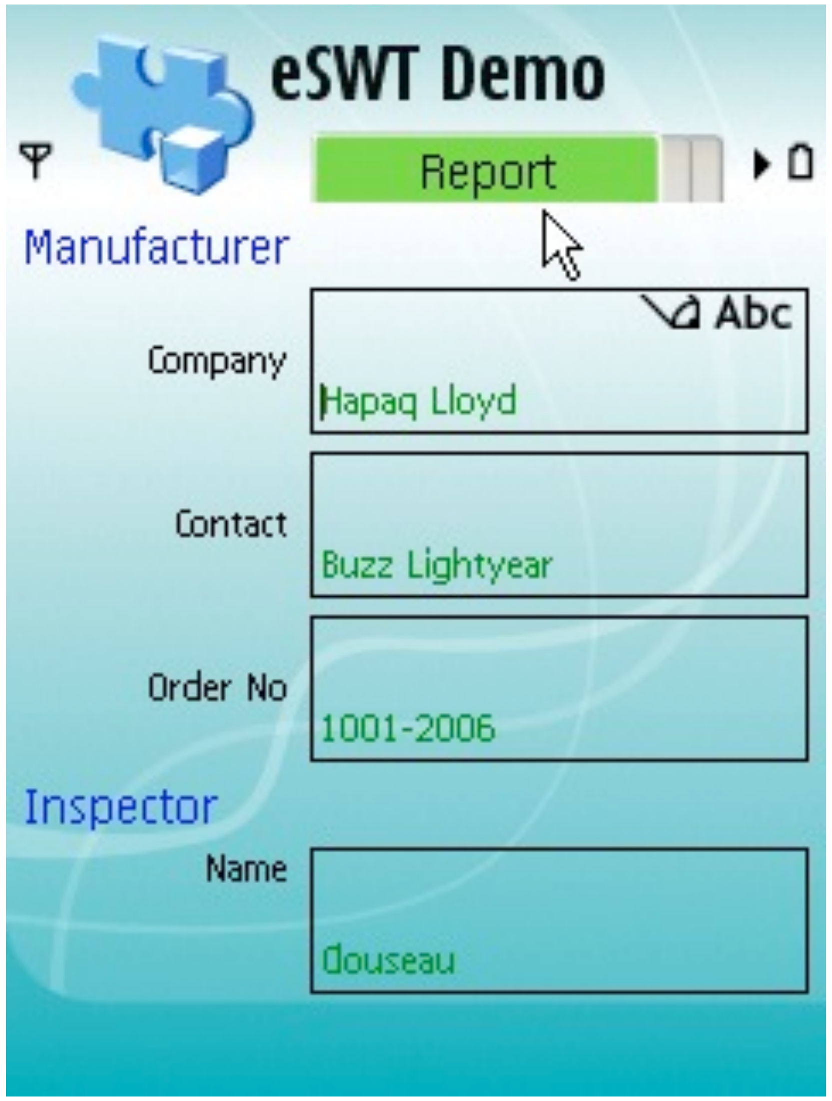
The Environment section contains the following environment and setup information:

- Environment settings including locale, processor architecture, operating system and windowing system.
- Java runtime environment.
- Program and VM arguments.
- Implicit plug-ins, which should always be taken into account when calculating dependencies.

Overview | Content | Environment

Embedded Rich Client Platform (RCP)

embedded Rich Client Platform: RCP meets device!

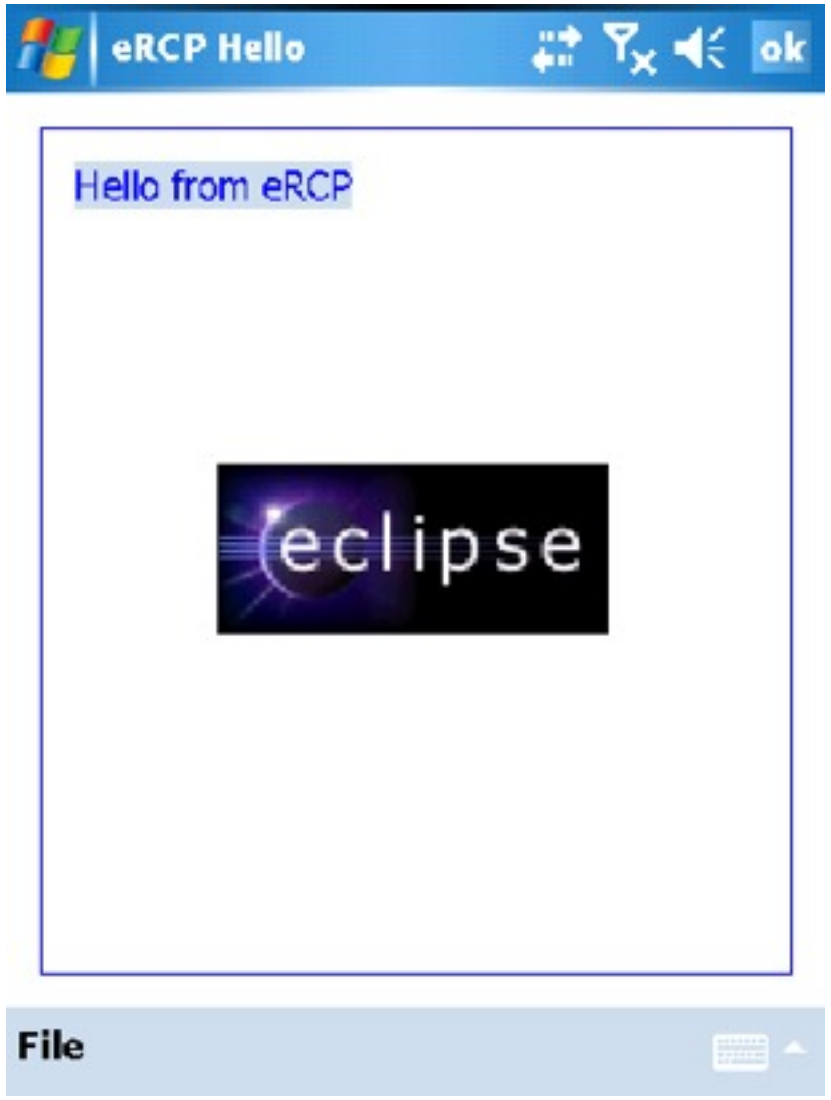


Manufacturer

Company	Hapaq Lloyd
Contact	Buzz Lightyear
Order No	1001-2006

Inspector

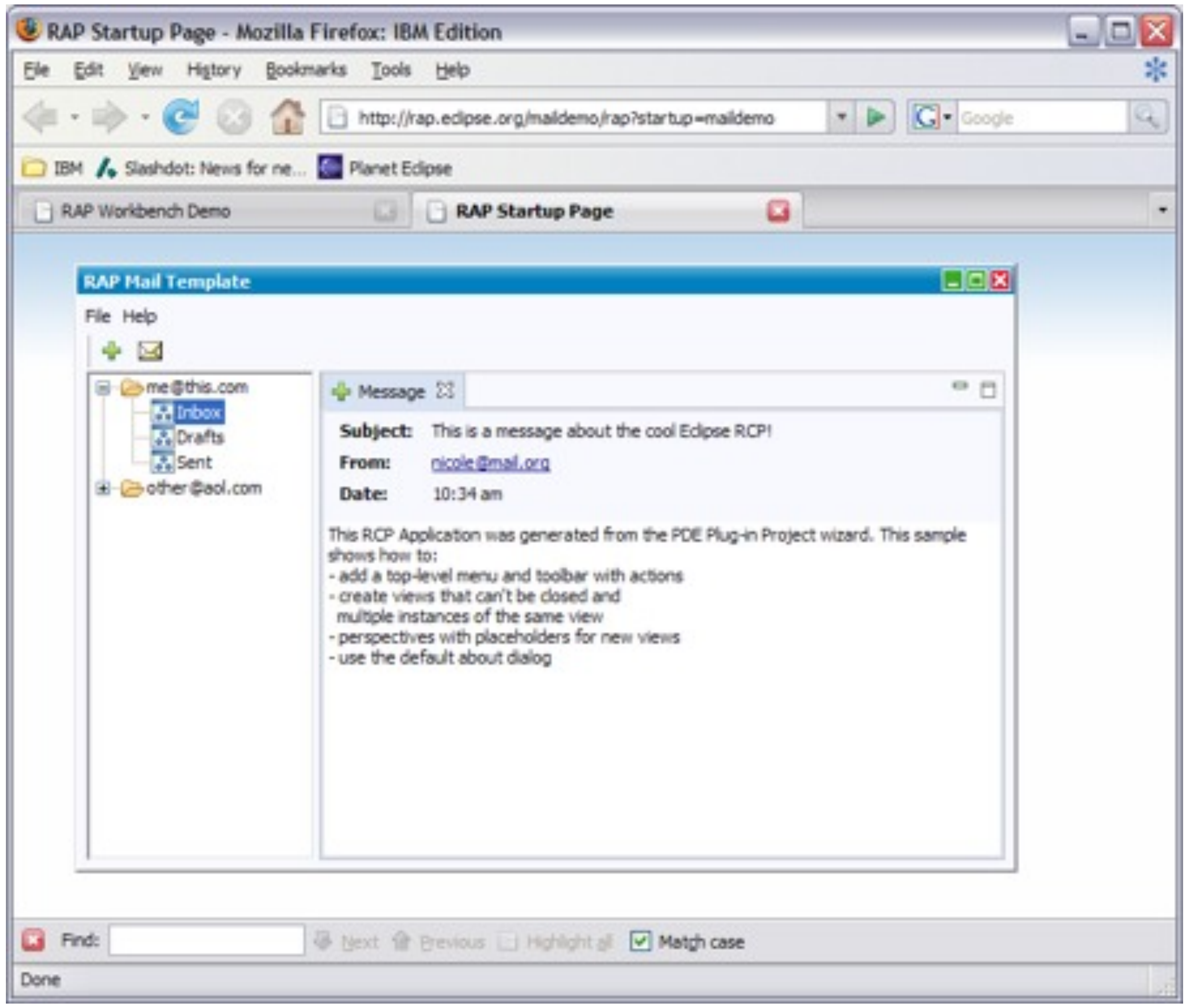
Name	Clouseau



Hello from eRCP

Rich Ajax Platform (RAP)

Rich Ajax Platform (RAP): RCP meets the Web!

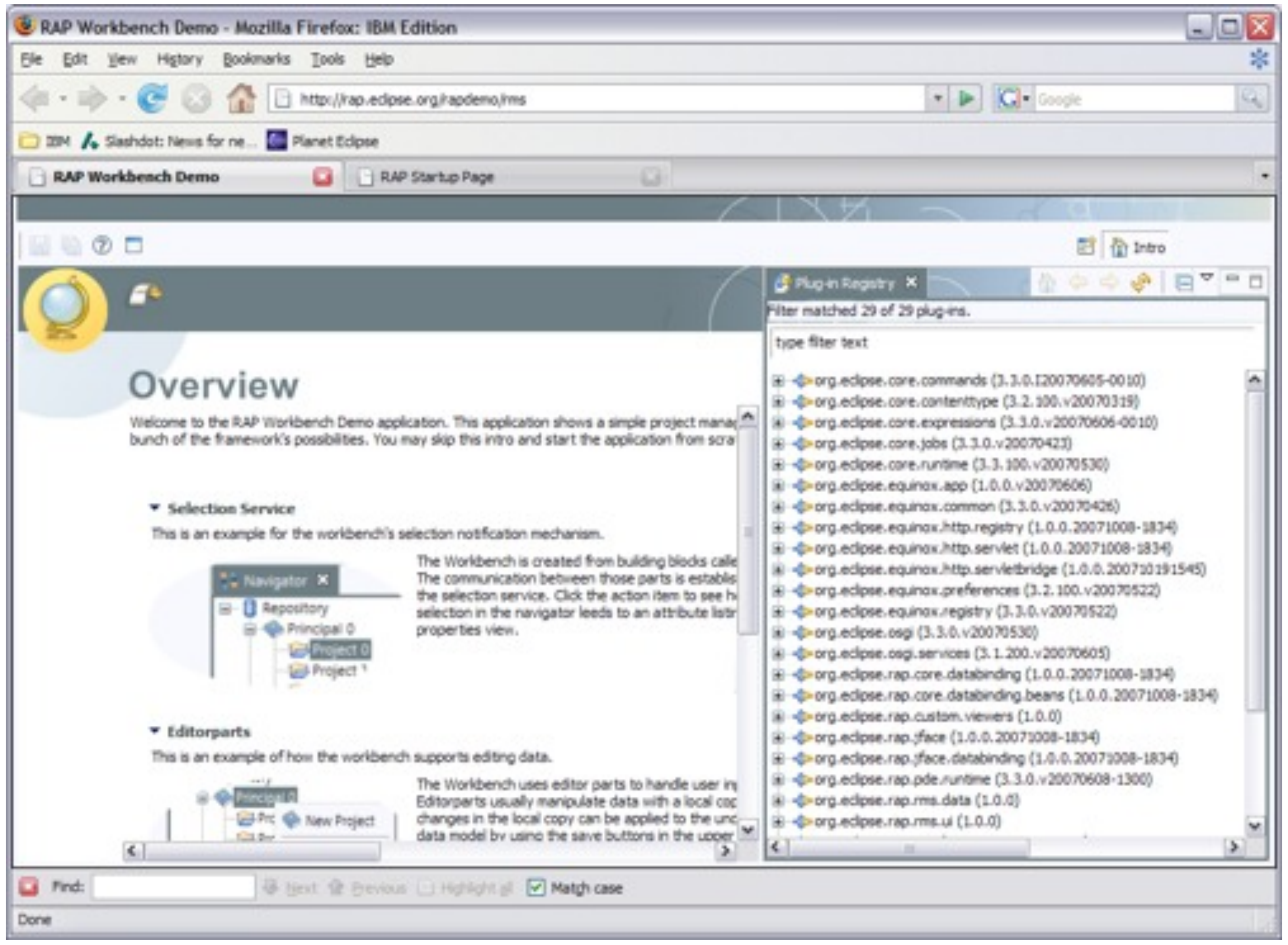


RAP Startup Page - Mozilla Firefox: IBM Edition

http://rap.eclipse.org/maildemo/rap?startup=maildemo

Subject: This is a message about the cool Eclipse RCP!
From: nicole@mail.org
Date: 10:34 am

This RCP Application was generated from the PDE Plug-in Project wizard. This sample shows how to:
- add a top-level menu and toolbar with actions
- create views that can't be closed and
  multiple instances of the same view
- perspectives with placeholders for new views
- use the default about dialog



RAP Workbench Demo - Mozilla Firefox: IBM Edition

http://rap.eclipse.org/rapdemo/rms

Overview

Welcome to the RAP Workbench Demo application. This application shows a simple project manag
bunch of the framework's possibilities. You may skip this intro and start the application from scra

▾ Selection Service

This is an example for the workbench's selection notification mechanism.

The Workbench is created from building blocks calle
The communication between those parts is establis
the selection service. Click the action item to see h
selection in the navigator leeds to an attribute listi
properties view.

▾ Editorparts

This is an example of how the workbench supports editing data.

The Workbench uses editor parts to handle user in
Editorparts usually manipulate data with a local cop
changes in the local copy can be applied to the unc
data model by using the save buttons in the upper

Filter matched 29 of 29 plug-ins.

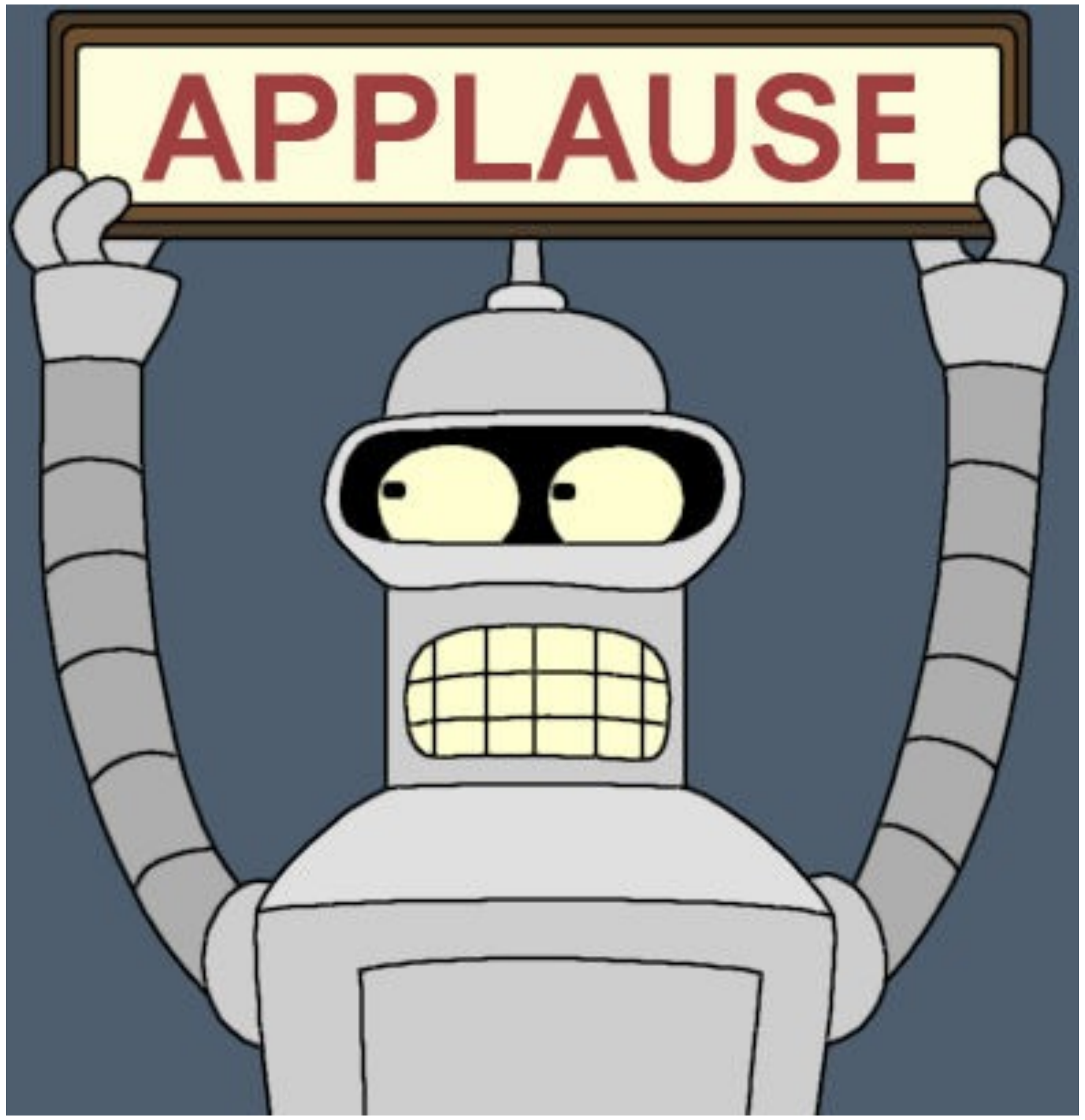
- org.eclipse.core.commands (3.3.0.I20070605-0010)
- org.eclipse.core.contenttype (3.2.100.v20070319)
- org.eclipse.core.expressions (3.3.0.v20070606-0010)
- org.eclipse.core.jobs (3.3.0.v20070423)
- org.eclipse.core.runtime (3.3.100.v20070530)
- org.eclipse.equinox.app (1.0.0.v20070606)
- org.eclipse.equinox.common (3.3.0.v20070426)
- org.eclipse.equinox.http.registry (1.0.0.20071008-1834)
- org.eclipse.equinox.http.servlet (1.0.0.20071008-1834)
- org.eclipse.equinox.http.servletbridge (1.0.0.200710191545)
- org.eclipse.equinox.preferences (3.2.100.v20070522)
- org.eclipse.equinox.registry (3.3.0.v20070522)
- org.eclipse.osgi (3.3.0.v20070530)
- org.eclipse.osgi.services (3.1.200.v20070605)
- org.eclipse.rap.core.databinding (1.0.0.20071008-1834)
- org.eclipse.rap.core.databinding.beans (1.0.0.20071008-1834)
- org.eclipse.rap.custom.viewers (1.0.0)
- org.eclipse.rap.jface (1.0.0.20071008-1834)
- org.eclipse.rap.jface.databinding (1.0.0.20071008-1834)
- org.eclipse.rap.pde.runtime (3.3.0.v20070608-1300)
- org.eclipse.rap.rms.data (1.0.0)
- org.eclipse.rap.rms.ui (1.0.0)

Agenda

- Plug-in Development with PDE
- Tips and Tricks
- Q&A

Conclusion

- http://www.eclipse.org/pde
- Want to contribute?
    - PDE Bug Day
    - http://wiki.eclipse.org/BugDay
- Thank you!



Questions?