



Plug-in Development Tips, Tricks and Best Practices

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

Chris Aniszczyk <zx@us.ibm.com>
PDE Lead
Eclipse Committer

October 14, 2009

Wednesday, October 14, 2009

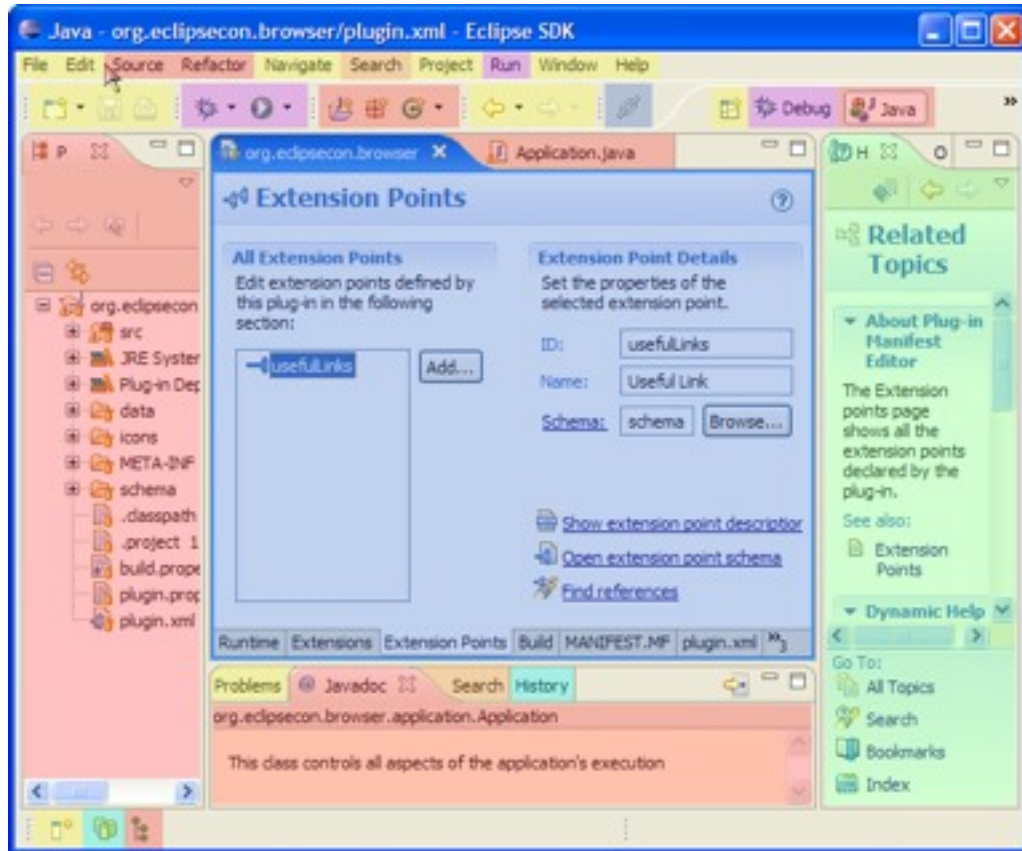
Agenda










-  **Plug-in Development with PDE**
-  **Tips, Tricks and Best Practices**
-  **Q&A**



Seamless Integration of Components



Component Legend

	PDE
	JDT
	User Assistance
	Debug
	Workbench
	Search
	Team

PDE



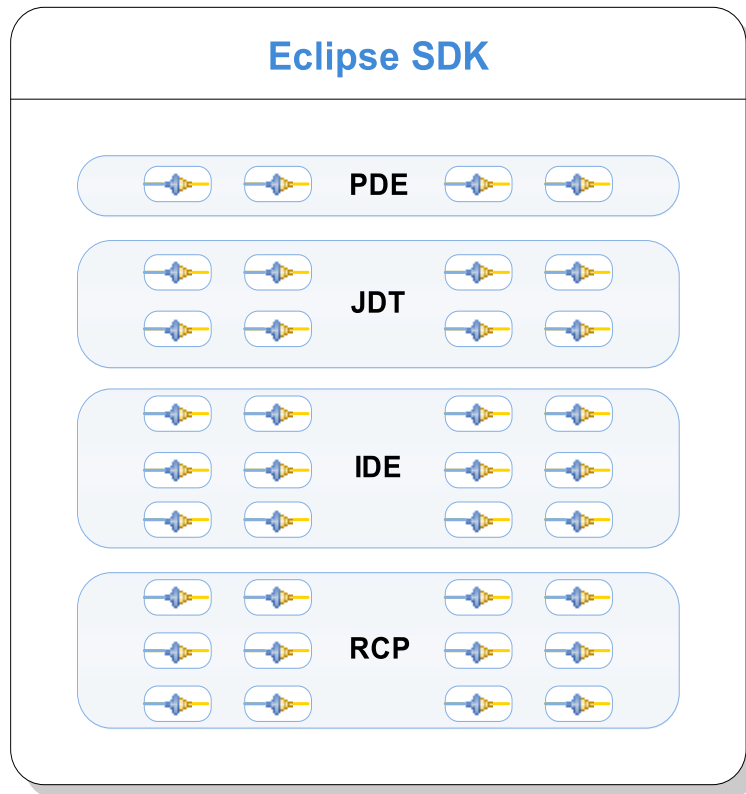
- PDE = Plug-in Development Environment
- Tools to develop Eclipse plug-ins
 - Wizards to create, import and export plug-ins and features
 - Specialized editors for plug-in manifest files
 - Templates for new plug-ins
 - Launchers to run, debug and test plug-ins
 - NLS tools to internationalize plug-ins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

Agenda



 **Plug-in Development with PDE**

 **Tips, Tricks and Best Practices**

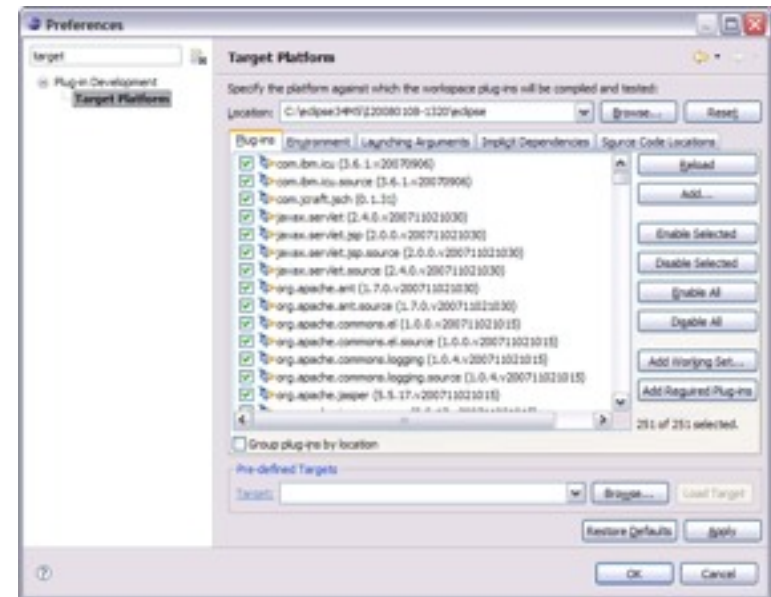
 **Q&A**



Target Management



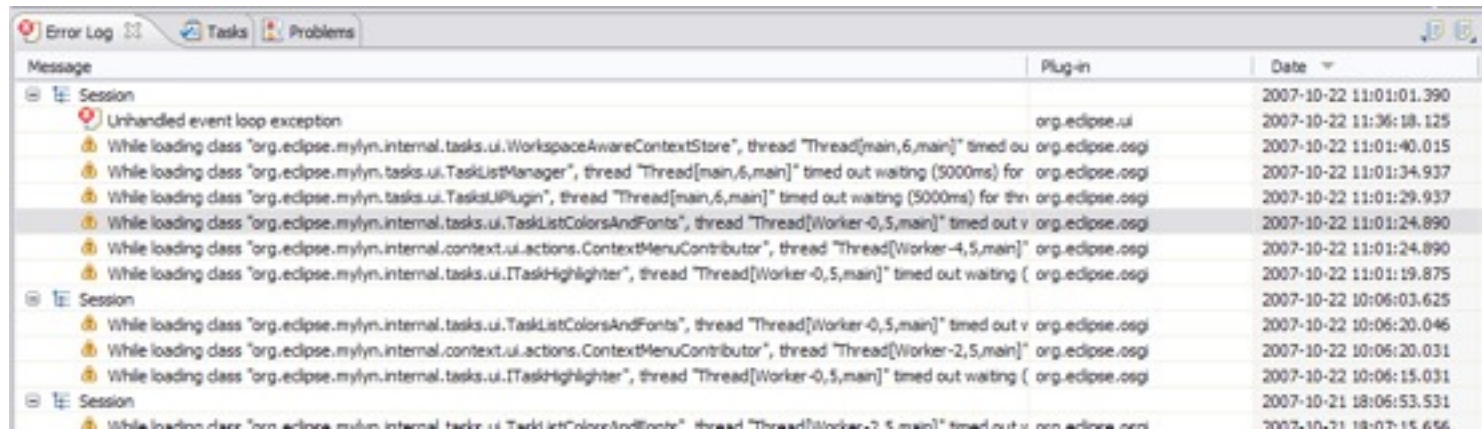
- Helps you to specify plug-in in which to build and run with.
- Includes tabs to set environment values, launching arguments, implicit dependencies, and source code locations.
- You can add plug-ins to the current target platform by using **target provisioners**. Current provisioners allow you to specify 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).



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 out waiting (5000ms) for the class loader to finish	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 the class loader to finish	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 the class loader to finish	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 waiting (5000ms) for the class loader to finish	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]" timed out waiting (5000ms) for the class loader to finish	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 (5000ms) for the class loader to finish	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 waiting (5000ms) for the class loader to finish	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]" timed out waiting (5000ms) for the class loader to finish	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 (5000ms) for the class loader to finish	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-0,5,main]" timed out waiting (5000ms) for the class loader to finish	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



OSGi 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/>



```
Console: M
hello [OSGi Framework] C:\Program Files\Java\jre1.5.0_08\bin\java.exe [Oct 21, 2009 10:00:00 AM]
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.

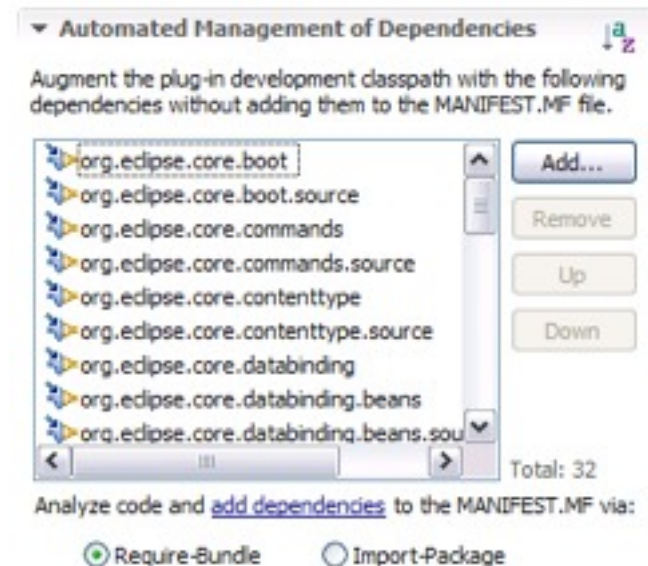
id      State      Bundle
0       ACTIVE     system.bundle_3.3.0.v20060919
1       ACTIVE     hello_1.0.0

osgi>
```


Automated Management of Dependencies



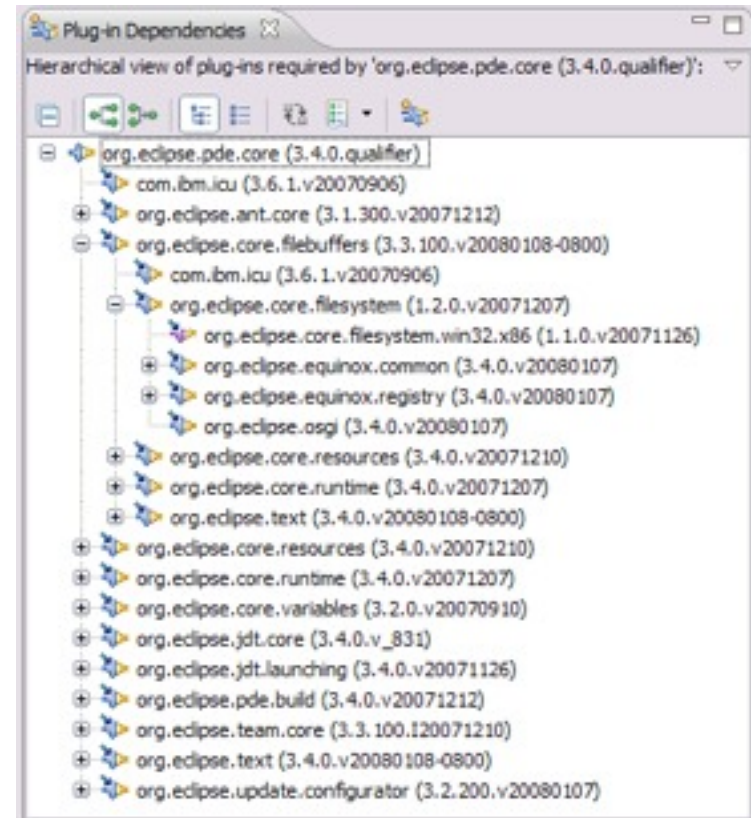
- Allows you to concentrate more on writing code than composing dependencies in your MANIFEST.MF.
- Helps **inexperienced** users formulate dependencies in their MANIFEST.MF so they can quickly create and run plug-in projects without in-depth knowledge of OSGi.
- **Update stale manifest files prior to launching** to ensure project will run correctly.



Plug-in Dependencies View



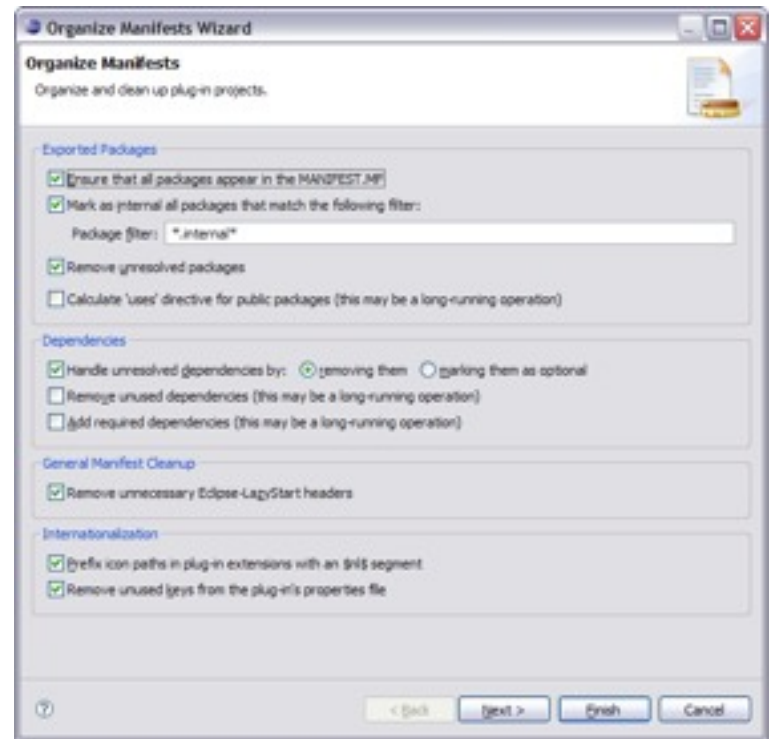
- The **Plug-in Dependencies** view allows you to see all the dependencies for any plug-in project.
- You not only see what plug-ins a project depends on, but also what plug-ins depend on that project (callers and callees).
- It also can display the current state, including dependency wiring, of the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



Organize Manifests Wizard



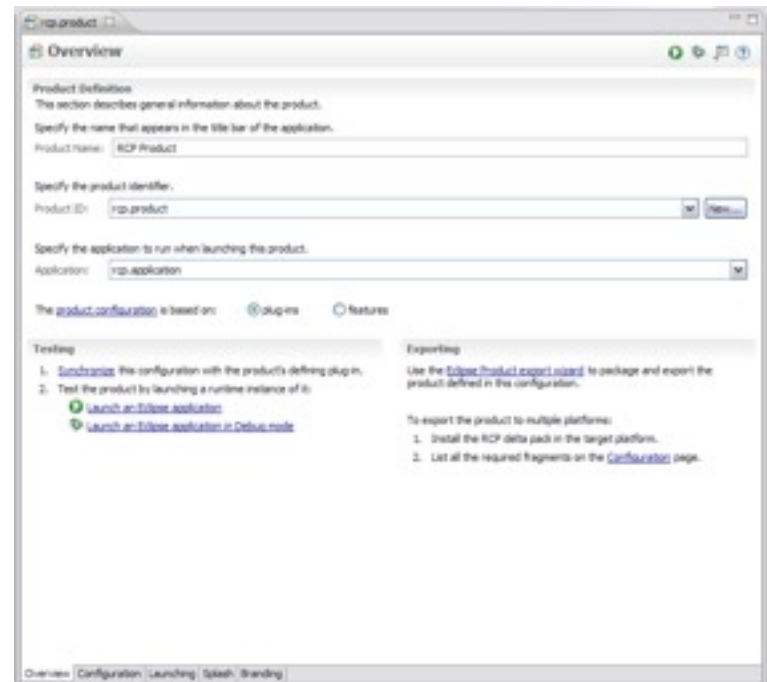
- Helps you modify and format your MANIFEST.MFs
- Accessible by right clicking on a MANIFEST.MF and selecting **PDE Tools > Organize Manifest**
- Helps you:
 - Export all packages in a project
 - Remove unresolved packages
 - Mark exported packages as internal
 - Modify unresolved dependencies (removing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



Product Editor



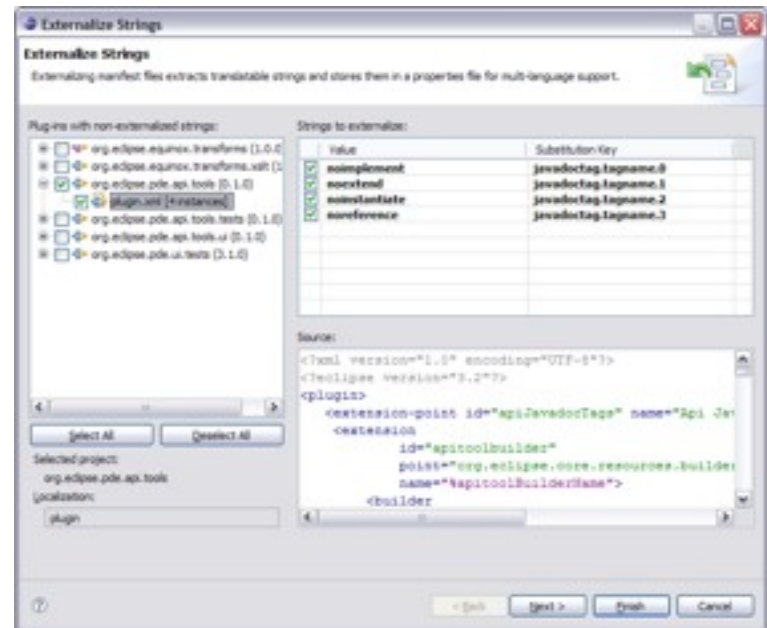
- A product definition helps you to easily customize, test and export an RCP/Eclipse application
- Customizations include:
 - Modifying which plug-ins are included
 - Create a splash screen
 - Bundling a JVM
 - Name for the launcher executable
 - Specify program and launching arguments
 - Define a welcome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



Externalization Strings Wizard



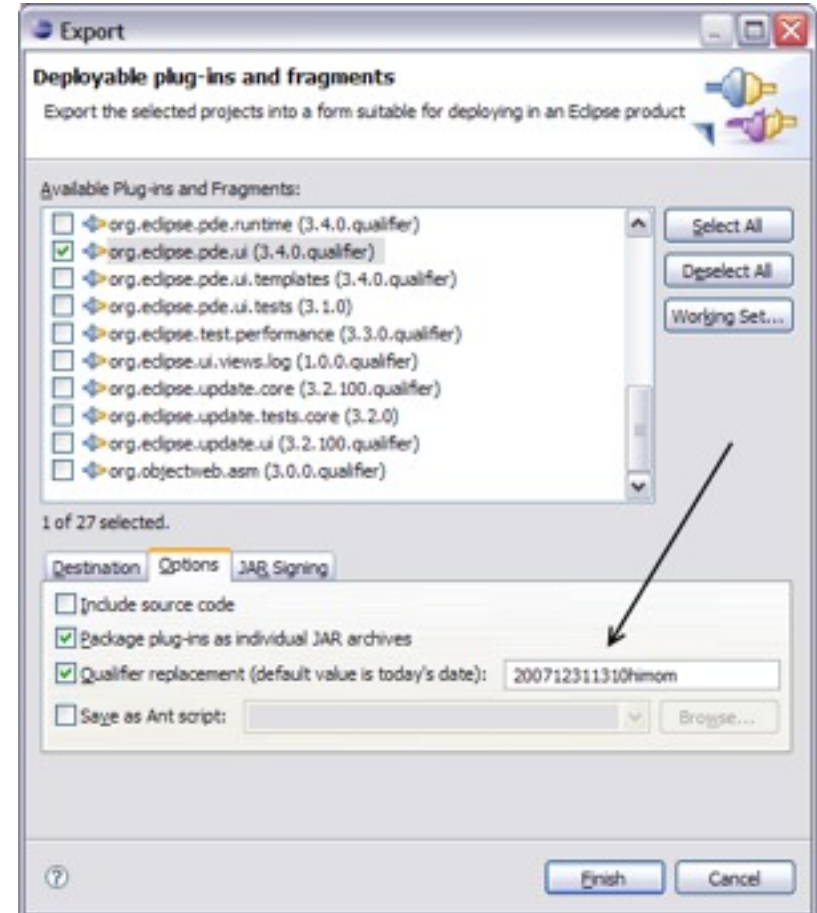
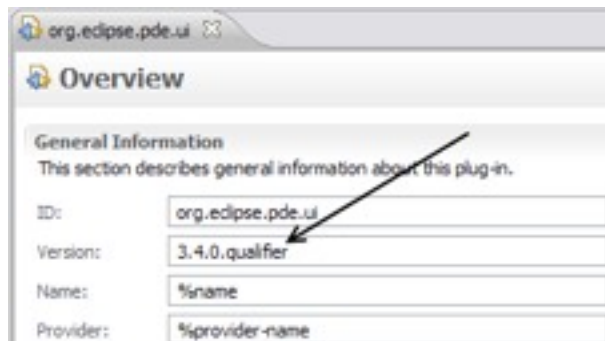
- PDE's Externalize Strings wizard allows you to quickly locate and painlessly externalize values in a plug-in's MANIFEST.MF and plugin.xml.
- Accessible by right clicking on a MANIFEST.MF or plugin.xml and selecting **PDE Tools > Externalize Strings...**
- Externalized values are put in a file specified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



.qualifier is awesome



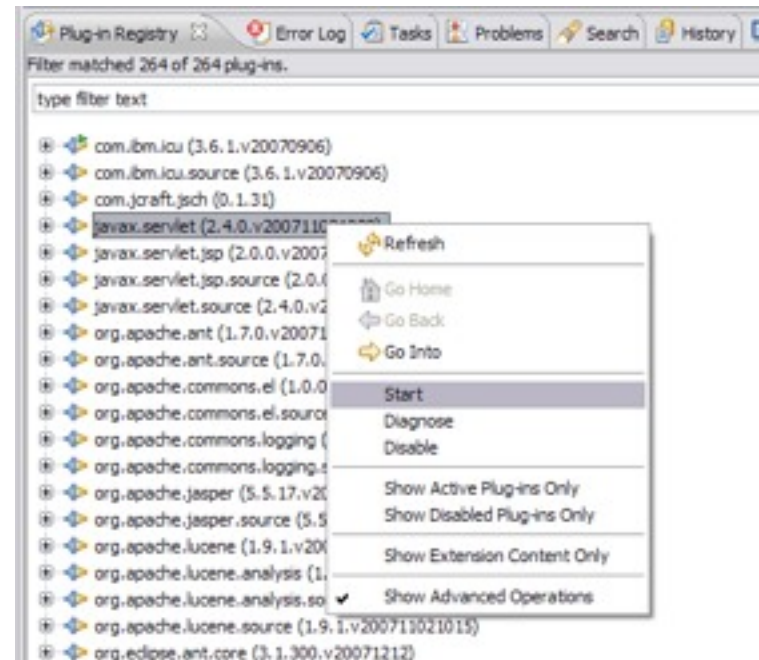
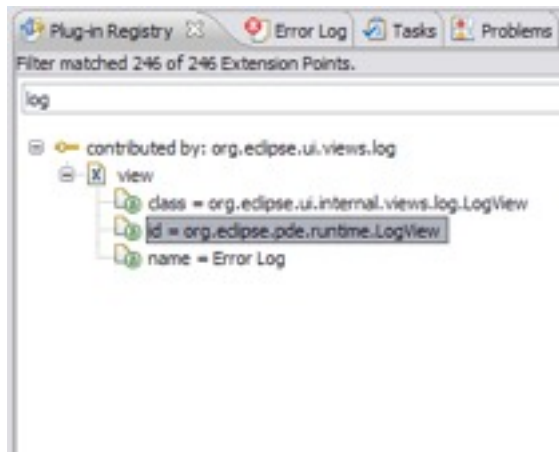
- The “.qualifier” marker allows you to easily substitute a value for the micro segment of a plug-in or feature’s version.
- The date is the default value, but you substitute any value when exporting your project using the PDE export wizards.



Plug-in Registry View



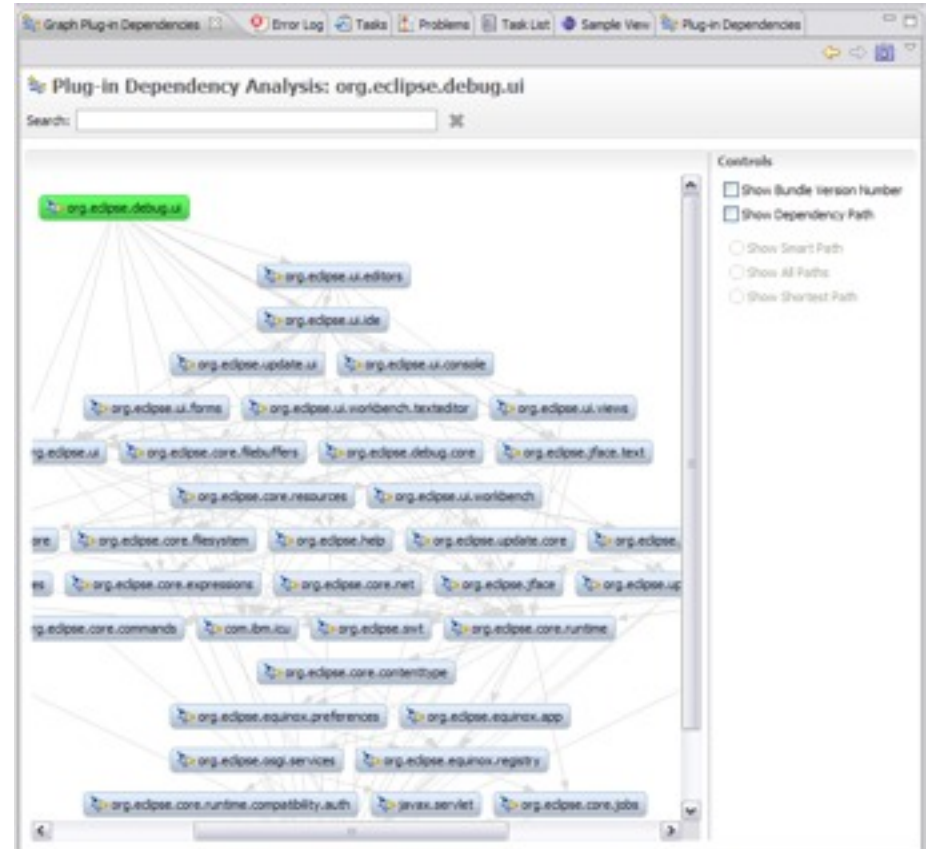
- The **Plug-in Registry** view is your eye into the runtime
- Show Advanced Operations
 - start/stop bundles
- Show Extension Content Only
 - quickly browse extensions



Graph Plug-in Dependencies View



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



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

Plug-in Project from existing jars



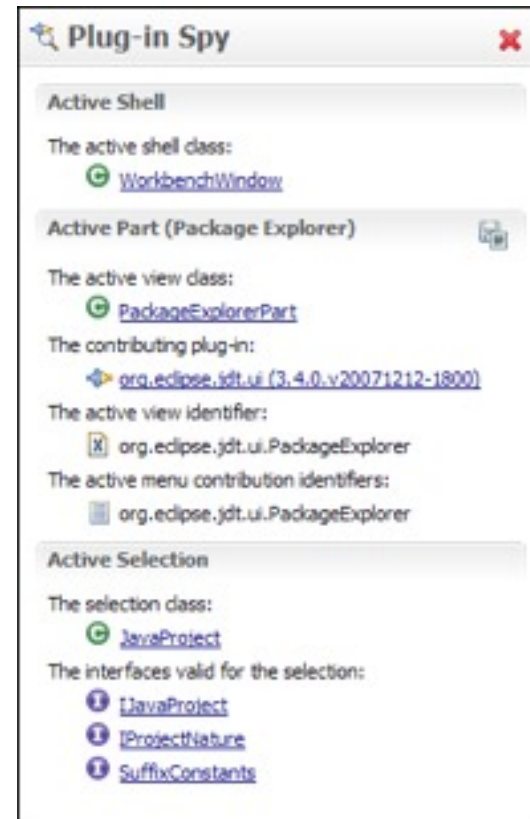
- The **Plug-in Project from Existing Jars** wizard enables you to quickly convert jar files to plug-ins.
- Helpful when an application is being converted to OSGi and it depends on certain library jars
- Can be very useful for utility jars, as they can be shared across multiple plug-ins instead of requiring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



Plug-in Spy (3.4M3)



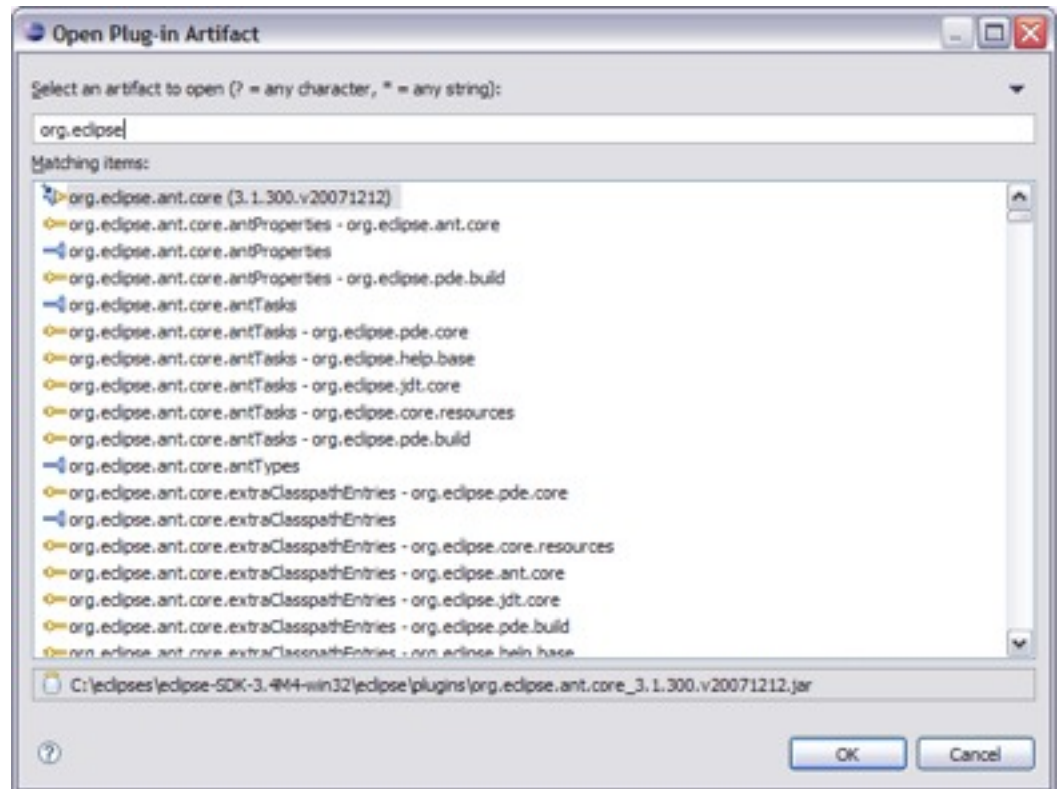
- ALT+SHIFT+F1
- Allows you to introspect what you're looking at...
- Hyperlinking
- Shows contributing plug-ins



Open Plug-in Artifact (3.4M4)



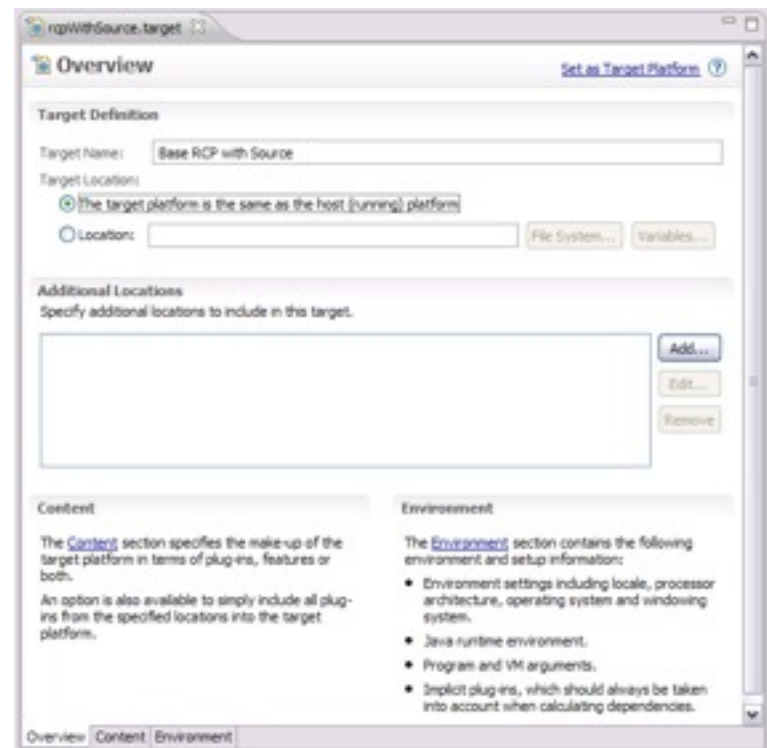
- Ctrl+Shift+A
- Quickly browse plug-ins, extensions and extension points



Target Editor



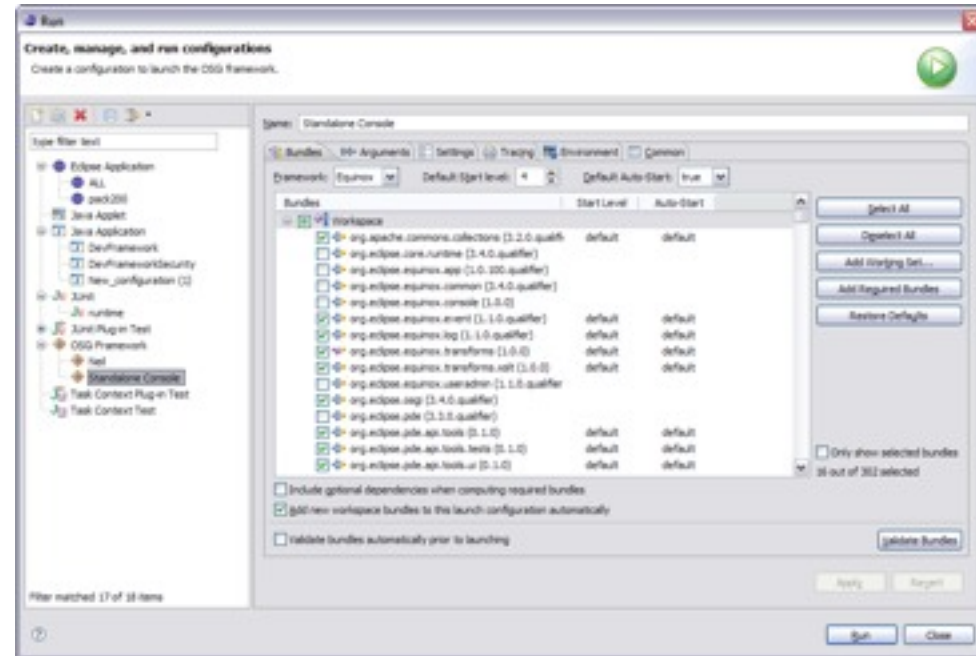
- A target definition is a file that helps to configure your PDE development environment.
- They can be created in the workspace or loaded from plug-ins who define them in your platform.



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.



Embedded Rich Client Platform (RCP)



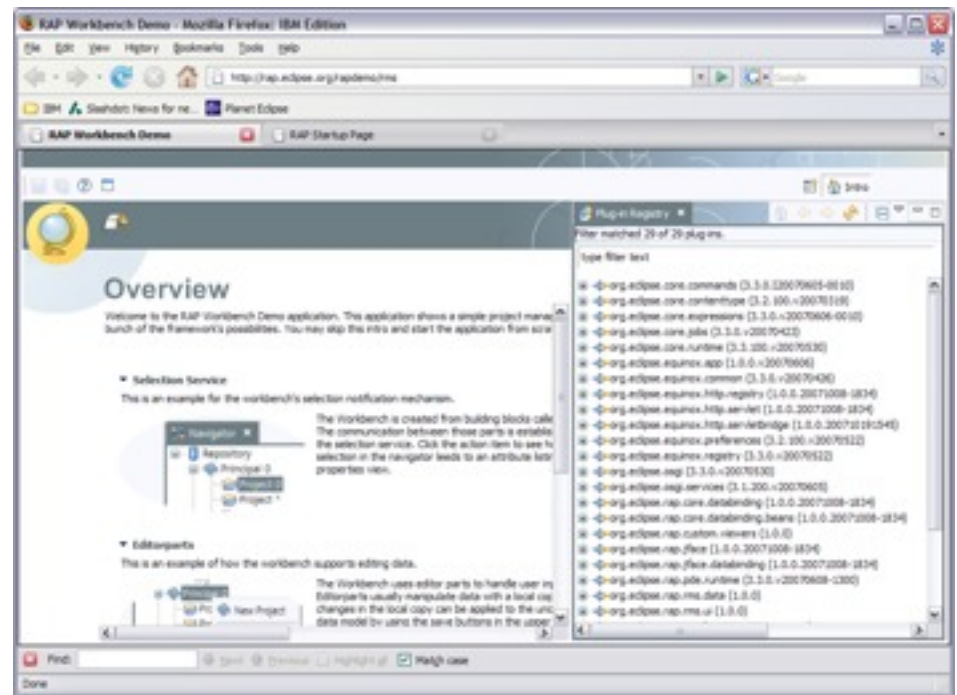
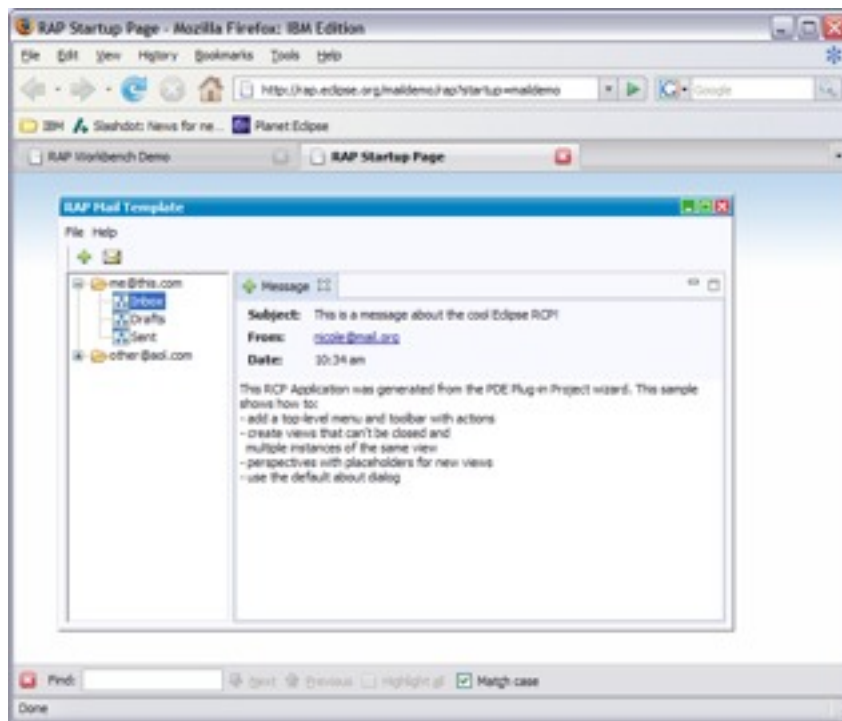
embedded Rich Client Platform: RCP meets device!



Rich Ajax Platform (RAP)



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



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

```
63 //  
64 * @see org.eclipse.core.runtime.IAdaptable#getAdapter()  
65 */  
66 public Object getAdapter(Class adapter) {  
67     if (adapter == IDebugElement.class) {  
68         return this;  
69     }  
70     if (adapter == IStepFilters.class) {  
71         return getDebugTarget();  
72     }  
73     if (adapter == IDebugTarget.class) {  
74         return getDebugTarget();  
75     }  
76     if (adapter == ITerminate.class) {  
77         return getDebugTarget();  
78     }  
79     if (adapter == IJavaDebugTarget.class) {  
80         return getJavaDebugTarget();  
81     }  
82 }
```

1 Items

Description	Resource
Illegally overrides org.eclipse.debug.core.model.DebugElement.getAdapter(Class)	JDIDebugElement

Agenda



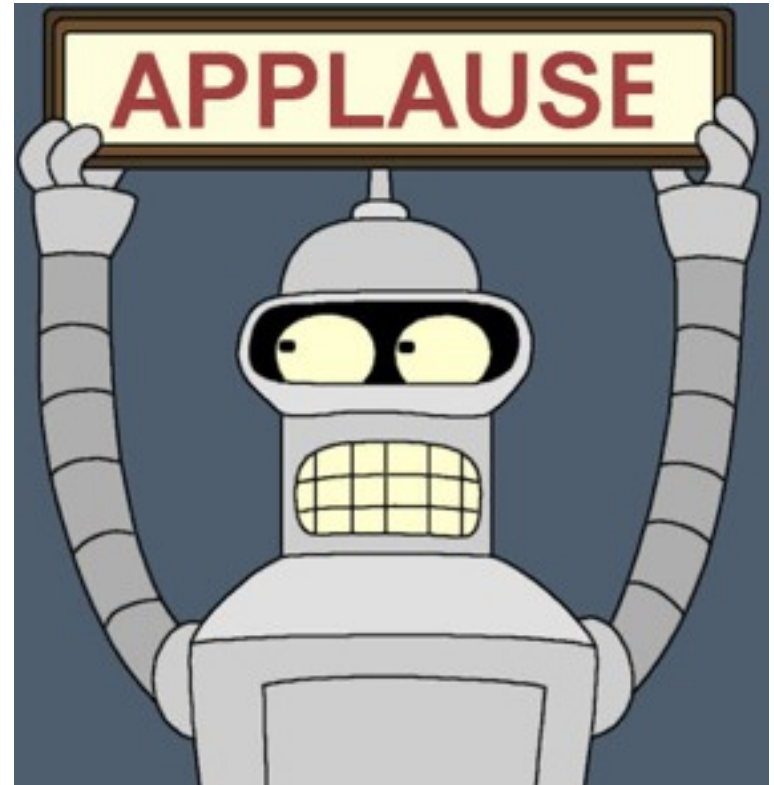
-  **Plug-in Development with PDE**
-  **Tips, Tricks and Best Practices**
-  **Q&A**



Conclusion



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





Questions?