

Plug-in Development 101 The Fundamentals

Chris Aniszczyk IBM Lotus, Austin

October 13, 2009

Tutorial Outline



The Basics
Anatomy of a Plug-in
Exercise One: The Eclipse Browser Plug-in
The Plug-in Manifest Editor
The Development Lifecycle of a Plug-in
Q&A

Tutorial Outline



- Anatomy of a Plug-in

- Exercise One: The Eclipse Browser Plug-in

- The Plug-in Manifest Editor

- The Development Lifecycle of a Plug-in

A89 🐠 –

What is Eclipse?



A very popular Java[™] IDE and much more...

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

What is Eclipse?



An open platform for anything and nothing in particular

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

An Open Platform



Eclipse is designed to be easily and infinitely extensible by third parties



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



IBM[©] Rational[©] Application Developer (RAD): An Integrated Development Environment (IDE)





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



Azureus: a Java BitTorrent client

Aceron	the last factor the				alD	12
10.00.01	1 0 0 01	3 6 6 6 6 9	*			
			~			
V Turrents Q	poons Statistics Phy Sh	are my buller 0.7511	Billian-20-0a-000-0mar			
General Peers []	Inam Peses Ples					
2	ded	T. Peces		South Speed	ta forest	1.
041111-167-160	Narine 3.4.2	L L L L L L L L L L L L L L L L L L L	30.7%	41.418/5	20.1 M/s	
05.228.239.190	Billionado 0.0.1	L DISCOULTER STATE	4.7%	41.718/5	8.3 M/s	
81.169.134.160	TorentPlan	L. Contraction of the second	100.0%	4.010(5	0.85	
94.70.207.170	Native 3.4.2	L CONTRACTOR	2.7%	2.4465	30.1 M/s	
网络新新	Natike 4.2.0	6	0.7%	1.146/s	0.8/s	
14.230.45.229	Billumado 0.3.7	A DESCRIPTION OF THE OWNER	6.75	28/5	0.85	
06-60-29-24	Native 4.6.2	L Designed and the second second	42.7%	189	0.6h	4
65.74.173.60	Uninova ([0	C LC	5.2%	184	0.8/4	
00.36.083.246	Marine 4.2.0	L DISCOURSE OF THE	29.75	0.00	0.8/4	
00.54.197.225	Approx 2.3.8.4	1 C	100.0%	7018/4	0.8/4	
24.118.6.100	Romes 2 3 8.4	L	100.0%	051.03	0.6%	
11.17.144.55	Routed 2.3.5.6	1	100.0%	0.00%	084	
94.00.1.004	Royma 2.3.5.6	L COLUMN TO A	30.7%	19.0%	0.8%	
80.28.66.77	Anres 23.8.4		15.9%	0.05	0.85	
303.217.70.209	Aneres 2.5.8.4		61.4%	185	0.85	
01.102.75.225	Anatus 2.3.8.4		100.0%	185	0.65	
01.461.151.715	Aneres 23.8.4		100.0%	185	0.8%	
62.41.25.166	Apres 23.8.4		100.0%	185	0.6%	
84.202.36.54	Approx 2.5.6.4		100.0%	185	0.8%	
90.144.41.247	Approx 2,2,5,6		100.0%	155	0.8%	
94.162.33.97	Rowey 23.8.6	L	100.0%	0.05	0.8%	
00.127.66.20	Bithomado-0.3.10	L .	40.9%	100	OBh	
111.099.002.03	Billionali-0.3.7		28.9%	184	oak	
12.107.123.207	ARC 3.5.1	L	24.9%	185	08%	
TEL INLUZIAN	MC24.9		41.7%	185	OBA	
298-228-232-246	Narine 3.4.2	6	41.7%	185	0.8h	1
ares 2.10.6	\varTheta 🥥 003,343 Usera 🛛 🕅	Rev 30, 17:525 (Per 0-0/1)1	D1 100.5 MB/H	U: [40K] 2	1.214(4)	1



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



IBM Lotus SameTime 7.5: a chat client





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



Games!: Sudoku

🗆 Eclipse Sudoku 📃 🗖 🔀								
File Sudoku								
8 8	0 🔛							_
	1				3		4	6
9		3		5				
				8			9	
	2		8	1			3	4
	8	4			5		1	
7	9	1	³4 _₿		4	2	6	5
					8			
8	4			6				
1	6	9						
1								



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



embedded Rich Client Platform: simple mobile applications



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



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





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

An Open Platform for Nothing in Particular



- No bias in the platform toward any particular domain or discipline
- Eclipse plug-in development is a level playing field



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Eclipse.org



An open source community that hosts over 60 open source projects



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



Downloading and Running the Eclipse SDK



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Supported Platforms





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Eclipse on Windows[™] XP



🗣 Java - org.eclips	econ.browser/plugin.xml - Eclip	ise SDK	
File Edit Source Ref	actor Navigate Search Project Ru *••••••••••••••••••••••••••••••••••••	n Window Help	× 6+4 € 24
Group edipsecon Solution Soluti	All Extension Points Edit extension points defined by this plug-in in the following section: Add Add Runtime Extensions Extension Points Problems @ Javadoc 22 Search arg.eclipsecon.browser.application.Appl	Extension Point Details Set the properties of the selected extension point. ID: useful.inks Name: Useful.Unk Schemat: schema Browse Schemat: schema	Related Topics About Plug-in Hamfest Editor The Extension points page shous all the extension points declared by the plug-in. See also: Dynamic Help Co To: Al Topics Search
K	This class controls all aspects of the a	oplication's execution	Bookmarks

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Eclipse on Windows Vista





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license



Eclipse on Linux[™]

d• 🖬 🛎 🔤	þ• 0•] <u>#</u> ₩ @•] ¢	• 0+ J#		討 参 Debug 劇	law
P 22 3 = D	👌 org echpseven browser 🖪 🚽	Application java	- 0	рн н о "	-
~	-4 ⁴ Extension Points		۲		
664	All Extension Points	Extension Point De	tails	Related Topic	s
♥ Sold org eclipsecon t ▷ Sold src	Edit extension points defined by this plug-in in the following section	Set the properties o selected extension	f the point.	About Plug-in Manifest Editor	
Þ 🛋 JRE System		ID: usefulL	inks	The Extension	
Plug-in Dep	-Qusetatoritis	Name: Useful I	Link	shows all the	
 D Conductors D Conductors 		Schema: schema	Browse	extension points declared by the plug-in.	
P Conscience P C		Show extension	point descript	Extension Points	
B plugin prope		40 Open extension	point schema	+ Dynamic Help	
aplugin.xml *		A Find references			•
splash.bmp	Extension Points Build MANIFES	T.MF plugin.xml ³⁰ 5		Go To:	
	Problems @ Javadoc 33 Sea	ch History	< * D	Search	
	org eclipsecon browser application Ap	plication		Bookmarks	
	This class controls all aspects of th	e application's executio	0	iii Index	

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Eclipse on Mac OSX





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Inside the Eclipse SDK





- RCP provides the architecture and frameworks to build any rich client application
- IDE is a tools platform and a rich client application in itself
- JDT is a complete Java IDE and a platform in itself
- PDE provides all the tools necessary to develop plug-ins and RCP applications

Rich Client Platform (RCP)





- Equinox is the runtime
- Standard Widget Toolkit (SWT) is a portable and native widget toolkit for Java
- JFace is a framework for common UI programming tasks
- Generic Workbench provides the UI personality of the Eclipse platform

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Integrated Development Environment (IDE)





- The IDE Workbench defines the Eclipse presentation
- IDE is an open tools platform:
 - Resource management
 - Text editing framework
 - A Language-independent debug model
 - Ant integration
 - Team repository integration
 - Help system
 - Update manager

Java Development Tools (JDT)





- JDT provides a complete Java IDE
- The compiler, which operates in incremental and batch modes, is also available as a separate download
- JDT is extensible:
 - Search and refactoring participants
 - Quick-Fix processors
 - Code Formatters
 - etc...

Plug-in Development Environment (PDE)





- PDE Does Plug-ins
- PDE Does RCP
- PDE Does Features and Update Sites
- PDE Does OSGi
- PDE Does User Assistance (as of Eclipse 3.3)

Plug-ins All the Way Down



E	Eclipse SDK				
	- PDE				
	JDT				
) IDE				
	RCP				

- 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

Layout of an Eclipse product





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Tutorial Outline



- The Basics



- Exercise One: The Eclipse Browser Plug-in

= The Plug-in Manifest Editor

- The Development Lifecycle of a Plug-in

A&Q =

A Fundamental Building Block





- A plug-in is a Java Archive (JAR)
- A plug-in is self-contained
 - houses the code and resources that it needs to run
- A plug-in is self-describing
 - who it is and what it contributes to the world
 - what it requires from the world

A Tale of Two Manifest Files





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

A Mechanism for Extensibility





Extension Point



Extension

- Extensibility in Eclipse is achieved via loose coupling
- Plug-in A exposes an extension point (the electric outlet)
- Plug-in B extends plug-in A by providing an extension (the plug) that fits into plug-in A's outlet
- Plug-in A knows nothing about plug-in B

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

If the Extension Fits...





- So many extension points...
- Each extension point is unique
- Each extension point declares a contract
- The extension point provider accepts only extensions that abide to the terms of its contract

A Declarative Approach





- Extension points and extensions are declared in the plugin.xml file
- The runtime is able to wire extensions to extension points and form an extension registry using XML markup alone

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Extensibility in Pictures





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Extensibility in Action



Preferences	
type filter text	Plug-in Development
 General Ant Help Install/Update Java Plug-in Development Run/Debug Team 	Set general plug-in development preferences. Show plug-in objects in editors and dialogs using: Identifiers Presentation names Automated Management of Dependencies Update stale manifest files prior to launching Destace Defendencies
0	OK Cancel

- Plug-ins may contribute preference pages
- All preference pages are assembled and categorized in the Preferences dialog
- How is the Preferences dialog created?
- How and when is a particular preference page created?

The Electric Outlet and the Plug





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license
Create the Preferences Dialog (1/3)



Preferences		
0	Ск	Cancel

- The UI plug-in provides the org.eclipse.ui.preferencePages extension point
- The UI plug-in first creates an empty Preferences dialog
- Now the dialog needs to be populated...

Generate the Preference Page Index (2/3)



Preferences	_ 🗆 🖾
type filter text General Ant Help Instal/Update Java Plug-in Development Run/Debug Team	
۲	OK Cancel

- The UI plug-in queries the extension registry for all org.eclipse.ui.preferencePages extensions
- The preference page index is then generated using the xml markup only:
 - Names for available preference pages are displayed in the tree using the name attribute
 - The category attribute is used to categorize the pages

Create the *Plug-in Development* Preference Page (3/3)

Preferences	
type filter text General Ant Help	Plug-in Development Image:
 Instal/Update Java Plug-in Development Run/Debug Team 	Identifiers Presentation names Automated Management of Dependencies Update stale manifest files prior to launching
0	Restore Defaults Apply OK Cancel

- When the *Plug-in Development* preference page gets selected, the UI plug-in asks the extension registry to load and instantiate the Java class specified by the class attribute of the corresponding extension
- The class gets loaded and the preference page gets created
- The plug-in providing that extension (i.e. the org.eclipse.pde.ui plugin) may then get activated, if it's not already active

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Tip of the Iceberg





- Plug-ins are connected without loading any of their code
- Code is loaded only when it is needed
- The lightweight declarative and lazy approach scales well
- An installed plug-in is not necessarily an active plug-in

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Declarative

Definition

Code

A Society of Plug-ins





- An Eclipse product is the sum of its constituent plug-ins
- Plug-ins are discovered upon Eclipse startup
- Plug-ins do not know how to play and interact with each other on their own

An Ordered Society of Plug-ins





- The Eclipse runtime manages all installed plug-ins and brings order and collaboration to their society
- A classpath for each plug-in is dynamically constructed based on the dependencies declared in its MANIFEST.MF file
- Every plug-in gets its own classloader

Unresolved Plug-ins





- If a plug-in has a dependency that is not met, the plug-in is deemed UNRESOLVED
- An unresolved plug-in does not get to interact with the rest of the plugins

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

A Chain Reaction





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Resolving the Unresolved





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Seamless Integration of Components



🗣 Java - org.eclips	econ.browser/plugin.xml - Eclipse SDK	
File Edit Source Ret	factor Navigate Search Project Run Window Help	
1 C3 - G1 @ 1 3	▶•••• (288 @• (\$••\$• #) (11 \$\$ 0ebu	a 🐉 Java 🎽 🇯
	🔐 org. edpsecon.browser 🗶 👔 Application.java 🖤 🗆	
C C C C C C C C C C C C C C C C C C	All Extension Points Image: Constant of the properties of the selected extension point. Edit extension points defined by this plug in in the following section: Image: Constant of the selected extension point. Image: Constant of the properties of the selected extension point. Image: Constant of the selected extension point. Image: Constant of the properties of the selected extension point. Image: Constant of the selected extension point. Image: Constant of the properties of the selected extension point. Image: Constant of the selected extension point. Image: Constant of the properties of the selected extension point. Image: Constant of the selected extension point. Image: Constant of the properties of the selected extension point descriptor. Image: Constant of the selected extension point descriptor. Image: Constant of the properties	Related Topics About Plug-in Hanifest Editor The Extension points page shows all the plug-in. See also: Extension Points Points Points
<	This class controls all aspects of the application's execution	ill Index
1 0° 🔞 😫		



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Tutorial Outline



The Basics
Anatomy of a Plug-in
Exercise One: The Eclipse Browser Plug-in
The Plug-in Manifest Editor
The Development Lifecycle of a Plug-in
Q&A

Exercises In the Form of Cheat Sheets



- Unzip both plug-ins into the "plugins" directory of your Eclipse installation
- Tutorial exercises can be accessed via Help > Cheat Sheets... from the main menu





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Exercise One: The Eclipse Browser Plug-in





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Exercise One: The End Result





- A view that shows an overview of the Eclipse project structure
- You can open associate web pages by clicking on nodes
- The view seamlessly integrates with the SDK

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Create the Eclipse Browser View



Create the Eclipse Browser view

- Introduction
- Open the Plug-in Development perspective
- Import the Edipse Browser plug-in

To import the sample plug-in for this tutorial, perform the following steps:

- Select "File->Import..." from the main menu to bring up the Import wizard. Expand the "Plug-in Development" category, and choose "Plug-ins and Fragments". Press "Next".
- Select the "Projects with source folders" radio button in the "Import As" section towards the bottom of the page. Press "Next".
- Select the "org.eclipse.browser" plug-in in the "Plug-ins and Fragments found" table. Click the "Add-->" button to move it to the table on the right . Press "Finish".
- ✓
 > Open the plug-in manifest editor
 ⑦

 ✓
 > Define a view extension
 ⑦

 ✓
 > Test the plug-in
 ⑦

- This exercise is structured as a 5step cheat sheet
- You use the plug-in import wizard to import the plug-in into the workspace
- You use the plug-in manifest editor to define the extension
- You use the Eclipse Application launcher to test the plug-in

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

2

?

1

Import the Eclipse Browser Plug-in



Import Plug-ins and Fragments	
Import Plug-ins and Fragments Create projects from plug-ins and fragments in the file system.	
Import From Type target platform (as specified in the Preferences) DrEctore DrEctore address 2.1 Locations to be searched for library JARs source attachment: Variables in library paths are resolved as specified in the Preferences: Plug-ins and Fragments to Import Splect from all plug-ins and fragments found at the specified location propert plug-ins and fragments required by existing workspace plug-in Import As Binary projects Binary	Target Platform
⑦ < Back Next >	Enish Cancel

- The plug-in import wizard brings a plug-in from the file system into the workspace
- The plug-in is converted from its deployed form (a JAR) to its development form (a workspace project)
- Choose to import the plug-in as "Project with source" if you wish to modify it.

Add a View Extension



New Extension	
Extension Point Selection	
Create a new Views extension.	=)
Extension Points Extension Wizards	
Extension Point filter: *.ul.	
- org.edipse.ul.viewActions	•
-{ org.eclipse.ul.views	
- org.edipse.ui.workingSets	-
information (like the workspace), ope editor. The user can make a view visi Available templates for views:	n an editor, or display properties for the active ble from the Window Show View menu or close it from 😿
Samele View	

To create a view to the workbench, you must extend the org.eclipse.ui.views extension point

All Extensions

Define extensions for this plug-in in the following section:

type filter text	
🗄 🗢 org.eclipse.browser.usefulLinks	
Image: org.eclipse.ui.perspectives	
org.eclipse.ui.views	

Define the Eclipse Browser View



II Extensions	Jaz 🖻	Extension Element Details		
Define extensions for this plug-in in t	the following	Set the properties of	"view". Required fields are denoted by "*".	
		id=:	org.edipse.browser.view	
type filter text		name*:	Edipse Browser	
🗈 🗢 org.edipse.browser.usefu	Li Add	dass*:	org.eclipse.browser.view.ui.EclipseBrowserView	Browse
Image: Org.eclipse.ui.perspectives Image: Org.eclipse.ui.views	Edit	category:		
Eclipse Browser (view)	Up	icon:	icons/eclipse_jcon.gif	Browse
	Down	fastViewWidthRatio:		

- The name and icon attributes are sufficient to put a placeholder for the view in the workbench
- \bullet The <code>class</code> is loaded only when the view is opened by the user

Test the Plug-in



Testing

Test this plug-in by launching a separate Eclipse application:

- Launch an Eclipse application
- 🎋 Launch an Eclipse application in Debug mode

- PDE launches a second Eclipse instance to show your plug-in in action
- Second instance uses a different workspace (i.e. a sandbox)



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Tutorial Outline



The Basics
 Anatomy of a Plug-in
 Exercise One: The Eclipse Browser Plug-in
 The Plug-in Manifest Editor
 The Development Lifecycle of a Plug-in
 Q&A

General Information



ID:	org.edipse.browser
Version:	1.0.0
Name:	%bundle.name
Provider:	
Platform Filter:	
Activator:	org.edipse.browser.view.BrowserPlugin

- A plug-in must have an ID, version and a name
- A platform filter is an optional field to specify under what conditions the plug-in should be allowed to run
- An activator controls the plug-in's lifecycle and may do initialization upon startup and cleaning up at shutdown

Execution Environment



Execution Environments	
Specify the minimum execution environ	ments required to run this plug-in:
➡ J2SE-1.4	Add
	Remove
	Up
	Down

Configure JRE associations...

Update the classpath settings

- An Execution Environment is the minimum JRE level required for a plug-in to run
- If a plug-in declares a J2SE-1.5 Execution Environment and Eclipse is running using a 1.4 JRE, the plug-in gets disabled gracefully

Dependencies

Required Plug-ins

Specify the list of plug-ins required for the operation of this plug-in:

org.eclipse.ui

org.eclipse.core.runtime

sorg.eclipse.ui.forms





- A plug-in must list all plug-ins that it needs to compile
- The runtime and development classpaths are computed based strictly on dependencies in the MANIFEST.MF
- PDE manages and updates the development classpath for you
- All plug-in dependencies must be met before a plug-in is resolved

Exported Packages



Exported Packages

Enumerate all the packages that this plug-in exposes to clients. All other packages will be hidden from clients at all times.

- org.eclipse.browser.application
- org.eclipse.browser.view
- org.eclipse.browser.view.model
- org.eclipse.browser.view.ui
- org.eclipse.browser.view.ui.actions
- org.eclipse.browser.view.ui.listeners
- org.eclipse.browser.view.ui.providers
- org.eclipse.browser.view.ui.sections
- org.eclipse.browser.view.util



- A plug-in may expose it code to downstream clients
- Downstream plug-ins may then make a dependency on the plug-in and use code from it

Extensions



II Extensions	Jªz [
efine extensions for this plug-in in the follow	ing section:
type filter text	
🕀 🗢 org.eclipse.browser.usefulLinks	Add
org.eclipse.ui.perspectives	Edit.
Eclipse.ui.views	
Cupse browser (view)	Up
	Down

- The Extensions section lists all the contributions the plug-in makes to Eclipse
- The plug-in manifest editor makes creating extensions easy because it is aware of the XML schema for all available extension points
- Hot links are available to jump back and forth between the manifest files and the source code

Extension Points



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

{usefulLinks

. - 212 matches

- org.eclipse.core.resources.modelProviders
- org.eclipse.core.resources.moveDeleteHook
- org.edipse.core.resources.natures
- -org.edipse.core.resources.refreshProviders
- org.eclipse.core.resources.teamHook
- org.eclipse.core.runtime.adapters
- org.eclipse.core.runtime.applications
- org.edipse.core.runtime.contentTypes
- org.edipse.core.runtime.preferences
- org.edipse.core.runtime.products
- org.eclipse.core.variables.dynamicVariables
- oro.eclipse.core.variables.valueVariables

- A plug-in may contribute 0 or more extension points to the platform
- The Eclipse SDK provides hundreds of extension points

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Add

Tutorial Outline



The Basics
Anatomy of a Plug-in
Exercise One: The Eclipse Browser Plug-in
The Plug-in Manifest Editor
The Development Lifecycle of a Plug-in
Q&A

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

From Genesis to Deployment





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Plug-in Creation



Plug-in Content Enter New Plug-in Project Templates Select one of the available templates to generate a fully-functioning plug-in. Plug Select one of the available templates to generate a fully-functioning plug-in. Plug Create a plug-in using one of the templates Available Templates: Available Templates: Plug Custom plug-in with a incremental projet Plug Select one of the auti-page entor Select Select one of the available templates: Plug Custom plug-in with a incremental projet Select Select one of the available templates: Plug Select one of the available templates: Plug-in with a propup menu Action sel. This template creates a simple Menu to the one on the button to the one on the button to the one on the button to the one on the one one on the on		łew Plug-in Project 🔄 📃 🔯
Enter New Plug-in Project Plug Select one of the available templates to generate a fully-functioning plug-in. Plug Select one of the available templates to generate a fully-functioning plug-in. Plug Create a plug-in using one of the templates Plug Coatom plug-in witard Plug Coatom plug-in with a incremental proje Plug-in with a incremental proje Plug-in with a neditor Plug-in with a popup menu Plug-in with a popup menu Plug-in with a popup menu Plug-in with a spopup menu Plug-in with a spopup menu Plug-in with a view Plug-in with a sample help content Mole the same Sample Action. Its role is to open a simple message of your choice. Extensions Used • org.eclipse.ul.actoriSets • org.eclipse.ul.actoriSets	Ph	g-in Content
Plug Select one of the available templates to generate a fully-functioning plug-in. Plug Image: Constant of the available templates to generate a fully-functioning plug-in. Plug Image: Constant of the templates Plug Image: Constant of the templates Plug Image: Constant plug-in with a more mental proje Plug Image: Constant plug-in with a incremental proje Image: Constant plug-in with a incremental proje Image: Constant plug-in with a incremental proje Image: Constant plug-in with a multi-page editor Image: Constant plug-in with a multi-page editor Image: Constant plug-in with a property page Image: Constant plug-in with a property page Image: Plug-in with a property page Plug-in with a sample help content Image: Plug-in with a sample help content Settersions Used Image: Plug-in with a multi-page editor Image: Plug-in with a multi-page editor Image: Plug-in with a property page Plug-in with a sample help content Image: Plug-in with a sample help content Image: Plug-in a stantistic plug-in are plug-in with a sample Image: Plug-in with a sample help content Image: Plug-in are plug-in p	1 *	🗧 🥥 New Plug-in Project 📃 🗖
Plug Plug Plug Available Templates: Plug Available Templates: Plug Second S	F	Select one of the available templates to generate a fully-functioning plug-in.
Clas Clas	F	up Create a plug-in using one of the templates Available Templates:
		 Custom plug-in wizard Custom plug-in wizard Custom plug-in wizard Plug-in with a incremental proje Plug-in with a multi-page editor Plug-in with a popup menu Plug-in with a property page Plug-in with a wiew Plug-in with sample help content Plug-in with sample help content Plug-in with sample help content Custom set. This template on the menu and the button invoke the same Sample Action set. This template on the menu bar and a button to the menu bar. Both the menu item in the new menu and the button invoke the same Sample Actions Used org-eclipse-ui.actionSets

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Life in the Workspace



Package Explorer 🛛 🗖 🗖
수 수 🗟 🗏 🖻 😓 🗸
🗆 🚮 org.eclipse.browser [dev.eclipse.org]
🖲 🔐 src
IRE System Library [IBM JDK 1.4.2]
🖲 🔜 Plug-in Dependencies
🖲 🗁 branding
🖲 🗁 data
🕀 🔄 icons
E META-INF
🖲 🗁 schema
🕀 🗁 solutions
.project 1.1 (ASCII -kkv)
build.properties 1.3 (ASCII -kkv)
plugin.properties 1.2 (ASCII -kkv)
plugin.xml 1.2 (ASCII -kkv)
splash.bmp 1.1 (Binary)

- The internal structure of a plug-in project in the workspace mirrors that of a deployed plug-in
- Two notable differences:
 - 1. The code is in source folders
 - 2. The plug-in project contains extra development metadata that are not part of the deployed plug-in

Editing the Plug-in





- The plug-in manifest editor is the central place to manage your plugin
- It provides hot links to
 - test and debug plug-ins
 - launch relevant wizards
 - quick navigation between source code and the manifest files

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Testing the Plug-in





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Configure the Build Content



Binary Build Select the folders and files to include in the binary build:

.project
🕀 🔽 🧁 META-INF
🕀 🔲 🗁 bin
🕀 🔽 🧁 branding
🔄 🔜 build.properties
🕀 🔽 🧁 data
🕀 🔽 🧁 icons
🛛 📄 plugin.properties
🖂 🕼 plugin.xml
🕀 🔽 🧁 schema
🕀 🔽 🧁 solutions
🛛 🗹 splash.bmp
🗄 🔄 🗁 src

- The plug-in project contains development-time metadata that should not be part of the deployed plug-in.
- On the Build page of the plug-in manifest editor, you check the list of files and folders that should be packaged

Externalize the Strings



Externalize Strings			160		6
Externalizing manifest files extracts translatal support.	ble strings	and stores them in a	properties file fo	or multi-language	8
Pug-ins with non-externalized strings:	Strings to	externalize:			
🐵 🛃 💠 org.edpse.brovser (1.0.0)	Value			Substitution Key	
Organization O	Ed	ipse Browser		view.name.0	
	6				(2)
	Sources				
¢ > >		<pre> class</pre>	"org.ecli "icons/ecl rg.eclipse "Eclipse B	pse.browser.v. ipse_icon.gif .browser.view rowser*/>	ev.1
Selected project: org.eclpse.browser Localization:	<td>çin></td> <td></td> <td></td> <td></td>	çin>			
plugin	¢		_		>

- PDE provides an *Externalize* Strings wizard that extracts translatable strings and stores them in a properties file for multilanguage support.
- This allows the plug-in manifest files to remain intact, while the properties files get translated

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Clean up the Manifests



🛛 Organize Manifests Wizard 📃 🗌 🔀
Organize Manifests
Organize and dean up plug-in projects.
Exported Packages
Insure that all packages appear in the MAVDFEST.MF
Mark as internal all packages that match the following filter:
Package (iter: "Jinternal"
Remove greesolved packages
Calculate \uses' directive for public packages (this may be a long-running operation)
Dependendes
Handle unresolved gependencies by: 🛞 removing them 🔘 garking them as options
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-LagyStart headers
Internationalization
Prefix icon paths in plug-in extensions with an \$nl\$ segment
Remove unused geys from the plug-in's properties file
⑦ Enish Cancel

- As the plug-in evolves, it may accumulate stale data
- The Organize Manifests wizard that inspects your code and manifests and removes or updates stale data

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license
Exporting the Plug-in



Export	
Deployable plug-ins and fragments	-0-
Export the selected projects into a form suitable for deploying in an Eclipse product	170
Available Plug-ins and Fragments:	
V sorg.edipse.browser (1.0.0)	Select All Deselect Al Working Set
1 out of 1 selected.	
Destination Options JAR Signing Orrectory:	
C:\Tools\Edipse\Browser	Browse
O Archive file:	
v	Browse
⑦ <back next=""> Finish</back>	Cancel

- The Plug-in Export wizard packages a plug-in into a deployable format
- Plug-ins can be exported en masse
- Plug-ins can be exported as an archive or as a directory structure

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

From Genesis to Deployment





Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Tuesday, October 13, 2009

Tutorial Outline



The Basics
Anatomy of a Plug-in
Exercise One: The Eclipse Browser Plug-in
The Plug-in Manifest Editor
The Development Lifecycle of a Plug-in
Q&A

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Tuesday, October 13, 2009

Questions and Answers?



Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license

Tuesday, October 13, 2009

Legal Notices



- Java 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.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- IBM, Lotus, Rational are copyrights of IBM Corporation in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.

Plug-in Development 101 | © 2008 by IBM; made available under Creative Commons Att. Nc Nd 2.5 license