

Introduction • I am going to show how to extend the MQ Explorer with new function - There will be code - But not too much • My experience ... SupportPac MS0P now contains 11 plugins providing - Event formatting - Activity monitoring - CSV exports for table - Connection wizard - Message Manager - Topic Viewer - Traceroute - Remote Admin See http://www.youtube.com/playlist?list=PLEE594DC49986AB67 - Search for MS0P © 2013 IBM Corporation





Plug-ins

- · Eclipse runtime is a small core program
- · Provides extension points
- · Plug-ins use those extension points
- · Runtime dynamically discovers the plug-ins





MQX packaging

- · Packaging has changed over time
- · Originally shipped only as part of the full MQ product
- Now also downloadable as SupportPac MS0T
 - Identical code to that in the product
 - But does not contain JNI libraries so cannot see any local queue managers
 - "Client-only" approach
 - Embeds the MQ Java client
- · Originally included a full Java development environment
 - Though not visible by default
 - One of the standard Eclipse packages
- · Now includes only the pieces needed for MQX to run
 - Runs as "RCP" application
 - But can be imported into other Eclipse environments
 - CICS Explorer, IDE etc
- Examples here will show V7.1/V7.5 formats

© 2013 IBM Corporation

Writing plug-ins

- An MQ plug-in exploits MQ-specific extension points
 Making it easy to access defined queue managers
- · But also any other Eclipse extension point
- There's no requirement for an MQX plug-in to do anything with MQ
 For example, the service definition repository is mostly independent
- · But you will probably use the MQ Java classes to interact with MQ
 - Full MQI available
 - PCF classes



What can you extend and mod	dify
 In the navigator view: Can add menu items to any tree node Can add new nodes 	Control Contro Control Control Control Control Control Co
 In the content view: Can add menu items to items in table Can add new views Arbitrary content 	Image: Status Image: Status
 Can inhibit some MQX operations eg Block a queue manager from being deleted 	Image: Sample: Sample
 Can add tabs to object property windows 	Autor A
Can add to import/export	Accounting monoming Accounting Accounting monoming Accounting monoming Accounting

What can you not (easily) extend or modify

- Cannot delete existing menu items
 Or reorder them
- Cannot add to toolbars
- Cannot access object properties
 Unless you use PCF commands
- · Cannot add a new top-level tree node
- · Cannot use existing text (translations) or icons
- The shipped core Eclipse plug-ins are limited scope
 - For example, do not have gef (graphical editors) or emf (model editors)
 - These can be added to MQX
 - Or you might require MQX to be run inside another Eclipse

Adding public plug-ins	
IBM WebSphere MQ Explorer (Installation2)	
File Edit Window Help	
C MQ Explorer - Nav O Help Contents	
Key Assist Ctrl+Shift+L Q ⊡ ⊕ IBM WebSpher Check for Updates	
🔁 😂 Queue Man 🛛 Instal New Software Sph	
Available Software Check the items that you wish to instal.	
Work with: Eclipse - http://download.eclipse.org/releases/helios Add Find more software by working with the "Available Software Stess" preferent Find more software by working with the "Available Software Stess" Add	. nces.
type filter text	
Name Version	
Beect All Deselect All Deselect All	v
© 2013 IBM Cor	rporation

Getting started

- To begin writing a plug-in, you need a development environment
- I was originally able to use the MQX-provided IDE
 But that disappeared in V7.1
- My system now consists of a standard Eclipse 3.6.1 download
 And then adding the MQX plug-ins to it
 - And any other plug-ins you care about (eg version control)
 - Lets me target V7.1 and later levels of MQX
- Open the plug-in development perspective and create a new project
 - This will create basic parts you will need for any Eclipse plug-in
 - Activator class and plugin.xml
 - An MQ plug-in has additional requirements

🖶 Plug-in Development - Eclipse SDK			_ 🗆 :
File Edit Source Refactor Navigate Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source	Search Project Run Window Help	□ × Debug	≫ Plug-in De 💮 MQ Explore
com.ibm.mq.explorer.ms0p.activity com.ibm.mq.explorer.ms0p.commo	Select a wizard Create a Plug-in Project		
Com.ibm.mq.explorer.ms0p.connect Com.ibm.mq.explorer.ms0p.csv Com.ibm.mq.explorer.ms0p.dmpmo	Wizards:		
e ⊖ com.bm.mq.explorer.ms0p.eetms e ⊖ com.bm.mq.explorer.ms0p.feature e ⊖ com.bm.mq.explorer.ms0p.metacg e ⊖ com.bm.mq.explorer.ms0p.pubgst e ⊖ com.bm.mq.explorer.ms0p.pubgst e ⊖ com.bm.mq.explorer.ms0p.templat e ⊖ com.bm.mq.explorer.ms0p.templat e ⊖ com.bm.mq.explorer.ms0p.tester			
e 🖶 com.ibm.mq.explorer.ms0p.topic/w e 😂 com.ibm.mq.explorer.ms0p.topic/w e 😂 com.ibm.mq.explorer.ms0p.update e 😂 com.ibm.mq.explorer.remote.admir e 😂 com.ibm.mq.explorer.remote.admir e 😂 com.ibm.mq.explorer.remote.admir	Bava Project Ward Project from Existing Ant Buildfie		
MAD Grand And And And And And And And And And A	? < Back Next > Fin	ish Cancel	Location Type
e ⊜ RTC e ⊜ Snippet	Warnings (4 items)		

Wizard-created template

Com.ibm.mq.explorer.mqtc.demo 🖄			-
Dependencies			0 🌣 🌮
Required Plug-ins Specify the ist of plug-ins required for the operatio of this plug-in. Secong.eclipse.core.runtme Cover Properties Total: 2 Very Properties Total: 2	Imported Pack Specify package without explicity	ages s on which this p identifying their EST.MF build.pr	Add Remov Propertie Total: 0
Tasks 🚺 Problems 🕸			
49 errors, 4 warnings, 0 others			
Description A Re	source Path	Location	Туре
Errors (49 items)			
Marnings (4 items)			
	Combining explorer inglocemo 13 So Dependencies Required Plug-ins Specify the lat of hug-ins required for the operate of this hug-in. So-org.eclipse.core.runtme Dependencies Coverview Dependencies Runtme Extensions Exter Total: 2 Total: 2 Overview Dependencies Runtme Extensions Exter Overview Dependencies Runtme Extensions Exter Overview Dependencies Runtme Extensions Re Overview Dependencies Runtme Extensions Re Overview Dependencies Runtme Extensions Extension Overview Dependencies Runtme Extensions Overview Dependencies Runtme Extensions Overview Dependencies Runtme Extensions Overview Dependencies Overview Dependencies	Combining explorer implications Combining explorer implicatio	

Dependencies
 The Manifest file shows which plug-ins you are dependent on This file is edited in the same set of tabs with plugin.xml
 Can define version prereqs in here as well Most MQ plug-ins will have a similar set of dependencies
Solution Dependencies
Required Plug-ins
Specify the list of plug-ins required for the operation of this plug-in.
to org.eclipse.ui
Recurse.core.runtime the com ibm mg explorer µi (7,1,0)
%⊳ com.ibm.mq.runtime
becom.ibm.mq.commonservices
♣ com.ibm.mq.explorer.core
vecom.iom.mq.explorer.msup.commonservices (7.2.0)
© 2013 IBM Corporation

The MQX Extension Points

- · There are several extension points
- The most important provided by MQX are
 - registerPlugin
 - addTreeNode
 - addContentPage
 - addImportExportSubcategory
- And the most commonly-used Eclipse extensions used in parallel are

 popupMenus
 - preferencePages
- Define which extensions are being used, and any parameters, in plugin.xml
 Eclipse knows the format of this file and can help fill it in

© 2013 IBM Corporation

registerPlugin extension point

- · Registers a plug-in with MQX to receive event notifications
 - Your class implements IExplorerNotify
 - The plugin.xml stanza names the plug-in, names the class, gives descriptions
- Notifications include
 - MQX initalised or closing
 - Your plug-in is enabled or disabled
 - Queue manager added or removed
- Eclipse Java editor can create the class with empty methods
 Showing TODO
- The MQX Preferences section "Enable Plug-ins" shows your plug-in
 - Disabling a plug-in in here will automatically remove its nodes and content pages

Name Authorization Service Tests Broker Explorer Cluster plug-in GSS QDepth GSS Queue Dashboard GSS Tonic Dashboard	Vendor IBM IBM GSS GSS	Description WebSphere MQ Explorer Authorization Service T WebSphere Message Broker Explorer plug-in to I WebSphere MQ Explorer Cluster Plug-in Grafi Software Solutions Openth Plura-in
Authorization Service Tests Broker Explorer Cluster plug-in GSS Quepth GSS Queue Dashboard GSS Tonic Dashboard	IBM IBM IBM GSS CSS	WebSphere MQ Explorer Authorization Service T WebSphere MQ Explorer Authorization Service T WebSphere MQ Explorer Cluster Plug-in Grall Software Solutions Openth Plug-in
Broker Explorer Cluster plug-in GSS QDepth GSS Queue Dashboard GSS Tonic Dashboard	IBM IBM IBM GSS	WebSphere Message Broker Explorer plug-in to I WebSphere MQ Explorer Cluster Plug-in Grail Software Solutions Openth Plug-in
Cluster plug-in GSS QDepth GSS Queue Dashboard GSS Topic Dashboard	IBM GSS GSS	WebSphere MQ Explorer Cluster Plug-in Grail Software Solutions ODenth Plug-in
GSS QDepth GSS Queue Dashboard GSS Topic Dashboard	GSS	Grail Software Solutions ODepth Plug-in
GSS Queue Dashboard	CSS	
GSS Topic Dashboard	033	Grail Software Solutions Queue Dashboard Plug-i
obb ropic bashboard	GSS	Grail Software Solutions Topic Dashboard Plug-in
Import and Export	IBM	WebSphere MQ Explorer Import and Export Plu
JMS Administration	IBM	WebSphere MQ Explorer JMS Administration Plu
JMS Administration Tests	IBM	WebSphere MQ Explorer JMS Administration Tes
JMS Admin Menus	IBM	WebSphere MQ Explorer JMS Administration Mer
Managed File Transfer	IBM	WebSphere MQ Explorer Managed File Transfer F
Menus	IBM	WebSphere MQ Explorer Menus Sample Plug-in
Messages	IBM	WebSphere MQ Explorer Message Plug-in
MQ Telemetry	IBM	WebSphere MQ Telemetry UI
MS0P: Activity Viewer	IBM	WebSphere MQ Explorer - Activity Viewer
MS0P: Connection Wizard	IBM	WebSphere MQ Explorer - Connection Wizard Pl
MS0P: CSV Table Exporter	IBM	WebSphere MQ Explorer - CSV Table Exporter P
less en les sur		
Select None		
		Restore Defaults Apply
	MIS Administration Tests JMS Administration Tests JMS Administration Tests JMS Administration Tests Menus Messages MQ Telemetry MSOP: Activity Viewer MSOP: Connection Wizard MSOP: Connection Wizard MSOP: Cost Table Exporter	JMS Administration Tests IBM JMS Administration Tests IBM Managed File Transfer IBM Menus IBM Messages IBM MQ Telemetry IBM MSOP: Connection Wizard IBM MSOP: Connection Wizard IBM MSOP: CSN Table Exporter IBM IBM SoP CSN Table Exporter IBM

Plug-in lifecycle
 Plug-ins have a lifecycle Enabled/disabled Started/stopped
 Eclipse will load your plug-in Activator is initialised (constructor), then "start" Then MQX "enables" plug-in MQX may not yet have completed its initialisation So you may not be able to do anything yet After "start" AND "enabled" AND "explorerInitialised" THEN you can start working
 Asynchronous operations and parallelism means these may not always happen in the same order
 Similarly at shutdown, there are several opportunities to clean up
One class could extend AbstractUIPlugin and implement IExplorerNotify – I prefer to have two classes ©2013 IBM Corporation

addTreeNode extension point

- Allows plug-ins to contribute tree nodes to the Navigator View
- Plug-in specifies a class implementing ITreeNodeFactory
 - Responsible for creating and adding tree nodes
- The Navigator view extends the Eclipse CommonNavigator class
 - So there may be helpful methods available
 - For example to select a node
- One parameter to the extension point is the plug-in ID
 - This must match the name used when the plug-in was registered



popupMenus extension point

- Eclipse extension point
 But mediated by MQX
- · Adds menu items to objects in tables or nodes in tree
- Menu visibility can depend on state – e.g. only connected Queue Managers
- The class you write has a selectionChanged() method
 - When an item is selected either in the content or navigator view
 - Table items are an MQExtObject class
 - Tree nodes return that object via the getObject method
- Can easily derive which queue manager is associated with an object - For example, if your menu is selected for a queue
- Then the run() method is invoked

© 2013 IBM Corporation

popupMenus extension point

- Getting the order of your menu items correct can be challenging
 - Eclipse seems to add them in reverse order to how they appear in the XML
 - menubarPath will usually be set to "additions"
- Adding submenus can be done but it is not obvious
- Cannot have one entry to add an item for both an object and a tree node
 - For example, adding same menu to the qmgr in the Navigator and in the Content table
 - Same class can be referenced but needs to clauses in plugin.xml

Menu definition in plugin.xml fragment

<pre><extension point="org.eclipse.ui.popupMenus"> <objectcontribution <="" adaptable="false" objectclass="com.ibm.mg.explorer.ui.extensions.TreeNode" pre=""></objectcontribution></extension></pre>
<pre>id="com.ibm.mq.explorer.remote.admin.popup.GotoQMgr"> <visibility></visibility></pre>
<and></and>
<pre><objectstate <="" td="" value="com.ibm.mq.explorer.treenode.qm."></objectstate></pre>
<pre><objectstate <="" td="" value="com.ibm.mq.explorer.remote.admin"></objectstate></pre>
<action< td=""></action<>
class="com.ibm.mq.explorer.ra.menuactions.MyActions"
enablesFor="1"
id="com.ibm.mq.explorer.ra.action.GotoQMgr"
label="Select Server"/>
© 2013 IBM Corporation

Pting in Development - my first.plugin - UM WebSphere MQ Fit Edit Nerkyde Sergin Brukt Bun Workdow (teb	
	a
Extension Point Selection Greate a new extension from the org.eclipse.ui.popupMenus extension point.	Extension Element Details Set the properties of "plugnotetis" Extension Set the properties of "plugnotetis"
Extension Points Extension Wizerds Available extension points: 	End Josephin In weight Jugan End Josephin End Jos
Available templates for ong-edges-uit-popupMenus: OPpup Menu Show only extension points from the required plug-ins	n Ponto Budi pigan.oni budi propertes
<back next=""> Enish Cancel</back>	



Selecting the	object	
Plag in Development Plag in Development Plag in Development Plag and the Development	nt my first plugin - 184 WebSphere MQ rdh Droket Ban Window Bolo Select antries: MQExtO MQExtODigett G MQExtObjectBase G MQExtObjectDaletedEvent G MQExtObjectDeletedEvent	
	Qualifier:	Pontis (Buld) plugn, xml buld, properties
1 1	OK Cancel	<u>9</u>
		© 2013 IBM Corp











Java Attiliba	te Editor		🕀 Implemented Interfaces Sele 📃 🖿
lava Class Create a new Jav	a class,		Ghoose interfaces: TACtionDel
Source Folder:	my.first.plugin/src	Browse	IActionDelegate
Package:	my.first.plugin	Bro <u>w</u> se	IACtionDelegate2
Enclosing type:		Bro <u>w</u> se	
Na <u>m</u> e:	MyAction		
Modifiers:	public C default C private C protected abstract final static		
Superclass:	java.lang.Object	Brows <u>e</u>	
Interfaces:	❶ org.eclipse.ui.IActionDelegate	<u>A</u> dd	Qualifier:
		Remove	
Which method stub	os would you like to create? public static void main(String[] args)		
	Constructors from superclass Inherited abstract methods		
			<u>A</u> dd OK Cancel
	Einish	Cancel	

Simple action implementation









Tips, Tricks and Techniques (1)

- I found the Java to be "easy", but getting the plugin.xml right was harder
 - Find another plug-in that seems to hook similar ways to what you want
 - The XML is accessible to review, even when the rest of the source is not
- Some values need to be the same in both plugin.xml and the Java code
- No easy way to do that with a single definition so use good naming conventions
- Insert plenty of trace and debug in the code
 - I use a trivial wrapper to put things to stdout so they appear in the console when launched as child of the development environment
- · Do long-running work in background threads
 - GUI can only be updated in main thread
 - SWT will throw exception if called from wrong thread

© 2013 IBM Corporation

Tips, Tricks and Techniques (2) - Java reflection

- There are lots of classes and methods in Eclipse ... not always easy to work out what to use
- Java reflection classes and methods can be very useful
 Even when there are no guarantees of ongoing compatibility
- My CSV plug-in adds an item to the toolbar of most MQX tables
 And then discovers the content of those tables
- There is no convenient mechanism in MQX to do any of that
- So the CSV code works first by
 - Hooking documented Eclipse methods to find all existing windows and parts
 - And asking to be notified if new windows and parts are created
 - If the MQ content view part exists or is created then add to the toolbar
- · Then, it uses Java reflection to recursively decode the contents of the part
 - For each field in the current part, look at its type, and if it seems promising, look at that subtype's fields
 - Until it finds a Table widget
 - At first, done under the debugger by hand to see if it worked at all

Tips, Tricks and Techniques (3) – Templates and skeletons

- · After writing a few plug-ins, a skeleton of core function simplifies the next
 - More function than the wizard-generated outlines
 - Gets basic operations running soon after your initial idea
- · A template will deal with the lifecycle flows
 - Allocate and cleanup resources at the right times
 - Debug
 - A single menu item that can be invoked
 - Handy utilities for formatting or verification etc
- · Eclipse wizards can help initial steps
 - But copy/paste is soon faster
- I'm now starting to move utility methods out of individual MS0P plug-ins
 - And into a single "internal" plug-in
 - So there aren't multiple similar copies
- · Consider having simple standalone programs to drive dialog panels
 - To help with constant redesign of SWT layouts
 - Much faster than relaunching the whole Eclipse and getting to the right place

© 2013 IBM Corporation

Samples

- · Samples provided with product
 - "simple" and "menus"
- · These are in the plugins directory under your MQX installation
 - com.ibm.mq.explorer.sample.menus_<version>.jar
 - com.ibm.mq.explorer.sample.simple_<version>.jar
- · The jar files can be expanded to show source
 - Java code
 - XML configuration

MQX Documentation

- Javadoc provided for MQX external interfaces
- In a jar file in the plugins directory

ActionFilterNames ContentPage	PREV PACKAGE NEXT PACKAGE	precated index <u>Heip</u> FRAME
Content LitleBar DragDropFile DragDropTreeNode ExplorerExtension	Package com.ibm.mq.	explorer.ui.extensions
ExplorerNotifyEvent IContentPageFactory IEvolorerAbstract/moortEvoort	Interface Summary	
IExplorerExternalDragDrop IExplorerExternalDragDrop	IContentPageFactory	Defines the interface to be provided by cla MQContent View
IExplorerNotifyAdapter IExplorerRuntimeImportExport	IExplorerAbstractImportExport	Interface to cover both <u>IExplorerImpor</u> <u>IExplorerRuntimeImportExport</u>
IExplorer TreeNodeDragDrop ImportExportId IMQExtObjectChangedListener	IExplorerExternalDragDrop	The interface that must be implemented if a an external (non-Explorer defined) data ty
IMQExtObjectDeletedListener IPropertyTabFactory	IExplorerImportExport	Deprecated. Superseded by IExplorerR
ITreeNodeFactory MQExtObject MQExtObjectChangedEvent	IExplorerNotify	Event interface for notification of explorer being initialised and closed.
MQExtObjectDeletedEvent MQQmgrExtObject	IExplorerRuntimeImportExport	Interface for contributing import and expos
MQSetExtObject ObjectId PreferenceId	IExplorerTreeNodeDragDrop	The interface that must be implemented if a the TreeNodeTransfer data type

Eclipse documentation

- Full set of Javadoc for Eclipse classes is in the SDK
- SWT help is at <u>http://www.eclipse.org/swt/</u>
 Including samples and snippets
- · Books:
 - I used "Java Developer's Guide to Eclipse" (D'Anjou, Fairbrother et al)
 - Plenty of newer ones now available
 - And specialised ones on EMF etc

Testing

- Initial (unit) testing can be done from within the development environment
- "Run configurations"
 - What to launch
 - Special configurations
 - Which JRE
 - Which other plug-ins
 - Clean workspace?
- · The same configurations are used for debug
 - For setting breakpoints, looking at variables etc
- But you then need to test with real installed MQX instances
 - I have seen examples of missing prereqs or bad version numbers
 - The prereq checking seems different from the IDE launch than runtime

Run conf	igurations		
Run Configurations Create, manage, and Create a configuration to lat Create a configuration Create a configuration to lat Create a configuration Create a co	run configurations unch an Eclpse application. Name: My plugins - MQ71 binaries Man Mar Arguments & Plug Municenare hata Run Configurations Create a configuration to launch as Eder	s - user trace	×
My plugins - MQ71 binarie My plugins - MQ71 msg r My plugins - MQ71 msg r My plugins - MQ71 msg r My plugins - no debug My plugins - no debug	Construction C	Name: My plugins - MQ71 binaries - user trace	n Defaut Auto-Start: Talse Select AI Deselect AI Add Working Set Add Requred Plug-ins Restore Defaults Only show selected 347 out of 745 selected Validate Plug-ins Adply_Revert
	3		Run Close

Compiling, Building, Shipping

- · Plug-ins can be delivered and installed in a variety of ways
- Each plug-in may be a subdirectory structure or a single jar
 - Compilation process must produce these
 - Eclipse integrates with ant tasks for building
- · 3 installation mechanisms
 - Put the plug-in tree into the "dropins" directory
 - Put plug-ins in a private directory and point to it via a file in the "links" directory
 Message Broker Explorer uses this
 - Package for use with the Eclipse "site" installer
 - > More complex to build these, assembling individual plug-in jars and having a feature
 - > But may be preferred, especially for centralised delivery
- · One issue has been seen with Windows
 - When UAC is active, changes using any of these mechanisms can require running Explorer once "with admin authority"
 - Once processed, do not need admin authority

```
Extract from ant build.xml
<property file="${basedir}\..\build properties.xml"/>
<path id="classpath.refid">
      <fileset dir="${eclipsePluginsHome}">
               <include name="com.ibm.mq.**\**\*.jar" />
               <include name="com.ibm.mq.*.jar" />
               <include name="org.*\*.jar" />
               <include name="org.eclipse.*.jar" />
       </fileset>
       <fileset dir="${mqPluginsHome}">
               <include name="com.ibm.mq.**\**\*.jar" />
               <include name="com.ibm.mq.*.jar" />
       </fileset>
       <fileset dir="${rasHome}">
               <include name="**\*.jar" />
       </fileset>
</path>
<pathconvert targetos="windows"
      property="classpath" refid="classpath.refid"/>
<target name="classes" depends="clean">
<javac srcdir="${basedir}\src" executable="${javac}"</pre>
      destdir="${buildClassesPath}" fork="no"
                                                optimize="true"
      debug="false">
      <classpath refid="classpath.refid" />
</javac>
</target>
```

Troubleshooting

- Exceptions are not always visible
 They are often propagated up to Eclipse and ignored
- But they may appear in the errorlog
- They will appear in the console when launched from within the development environment
- Normal MQ error logs and FFSTs will be created if you misuse MQ

© 2013 IBM Corporation

Summary

- Writing plug-ins is a good way to add value to MQ
- Without needing to write a load of framework code
- Can get straight into the real work