

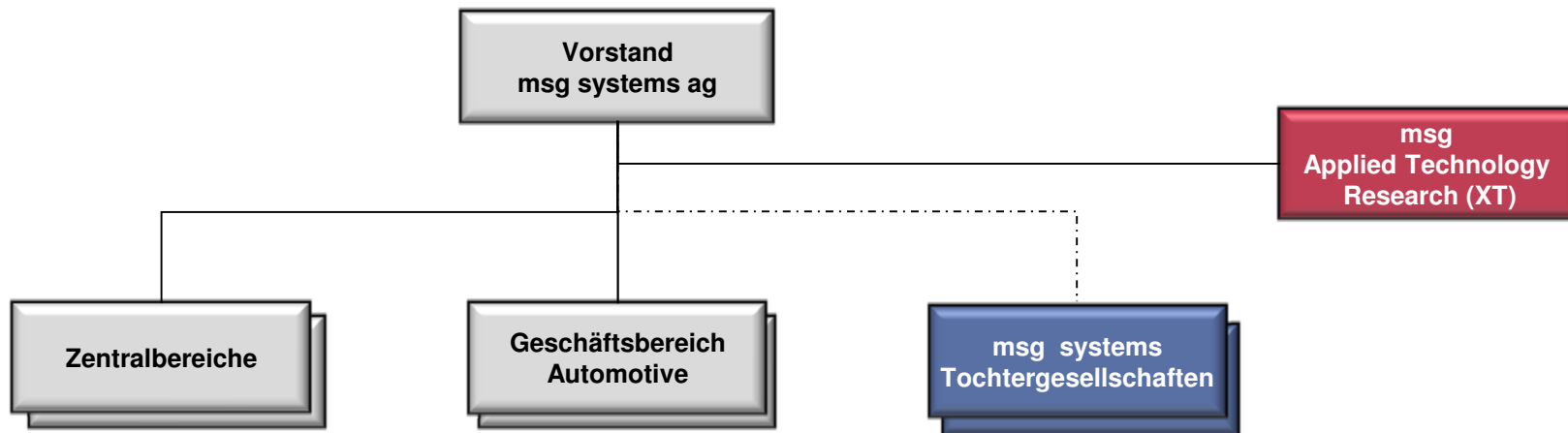
Eclipse RCP Part I

Automotive Financial Services Insurance Life Science & Healthcare Public Sector
Telecommunications & Media Travel & Logistics Utilities Automotive Financial
Services Insurance Life Science & Healthcare Public Sector Telecommunications
& Media Travel & Logistics Utilities Automotive Financial Services Insurance
Life Science & Healthcare Public Sector Telecommunications & Media Travel
& Logistics Utilities Automotive Financial Services Insurance Life Science
Healthcare Public Sector Telecommunications & Media Travel & Logistics
Utilities Automotive Financial Services Life Science & Healthcare Public
Sector Telecommunications & Media Travel & Logistics Utilities Automotive
Financial Services Insurance Life Science & Healthcare Public Sector
Telecommunications & Media Travel & Logistics Utilities Automotive
Financial Services Insurance Life Science & Healthcare Telecommunications
& Media Travel & Logistics Utilities Automotive Financial Services Insurance
Life Science & Healthcare Public Sector Telecommunications & Media Travel &
Logistics Utilities Automotive Financial Services Insurance Life Science &
Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities
Automotive Financial Services Insurance Life Science & Healthcare Public Sector



.consulting .solutions .partnership

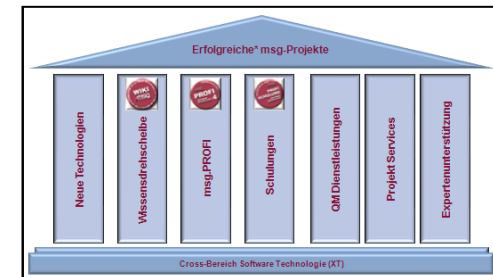
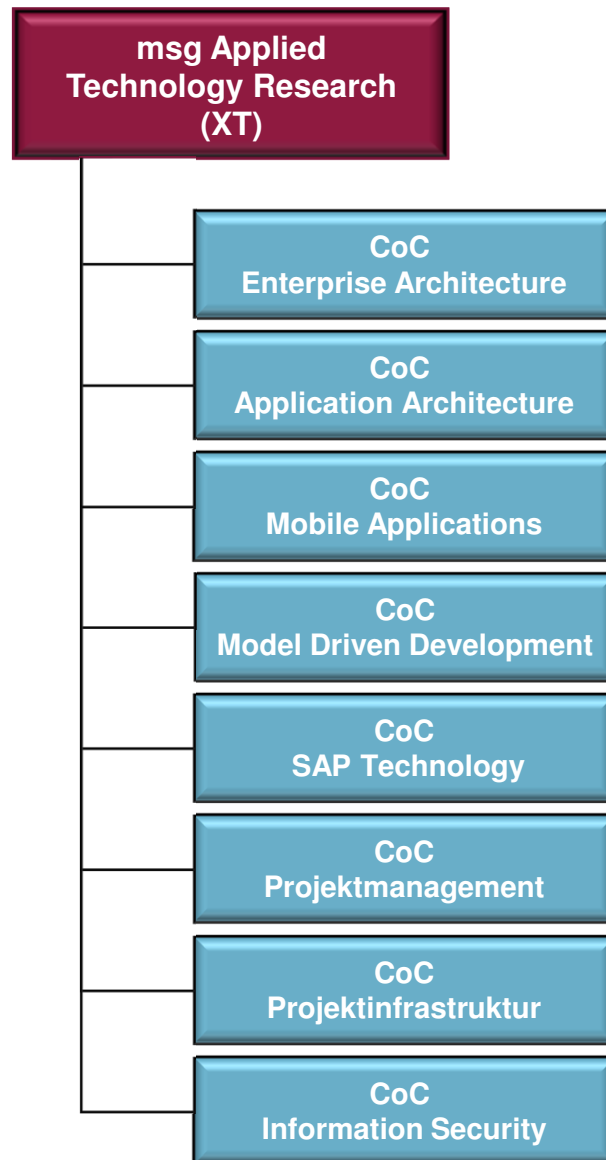




Der Crossbereich msg Applied Technology Research (XT) leistet einen wesentlichen Beitrag zum nachhaltigen Erfolg von Projekten der msg systems group durch:

- Etablieren von technischen und methodischen Innovationen
- Wissenstransfer und Ausbildung von Mitarbeitern
- Vereinbarung und Sicherstellen der Einhaltung von Standards
- direkte Unterstützung von Projekten

XT – Expertenunterstützung in definierten Themen

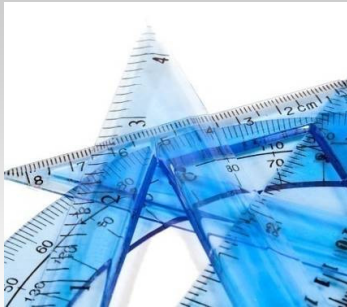


- **Aktive Projektarbeit**
 - Projektmitarbeit, Problemlösung, Review
 - PreSales-Unterstützung
- **Wissensvermittlung**
 - Training-on-the-job
 - Workshops, Schulungen
- **Coaching / Sparring**
 - Keyplayer im Projekt
 - Angebotsteam
- **Gestaltung / Verantwortung für:**
 - msg.PROFI <http://profi.msg.de>
 - msg.WIKI <http://wiki.msg.de>
 - msg.Techportal <http://techportal.msg.de>

XT adressiert alle wesentlichen IT-Aspekte Ihrer Projekte.



Software Engineering



Projekt- und Qualitätsmanagement



Dienstleistungen:

- Projektverantwortung
- Ergebnisverantwortung
- Audits, Reviews
- Training, Coaching



Architektur & Technologien



IT-Security

Welcome to the Eclipse RCP Workshop

who you are....

- Your Name?
- Java experience?
- Experience with Eclipse as IDE?
- Experience with Eclipse as a platform for rich client applications?
- Other technical background?

- Studium der Informatik an der TU-München
- 2003-2007 Siemens Corporate Research
 - Forschung und Entwicklung für den Software-Entwicklungs-Prozess
 - Tools auf Basis der Eclipse Rich Client Platform
- seit 2007 in der msg
 - Client-Architekt im GB Automotive für verschiedene RCP-Projekte
 - EPDM (Daimler)
 - ACM (Daimler)
 - BNE (BMW)
 - AVx (Audi)
 - Architekt / Trainer im CoC “Application Architecture” und “Mobile Applications” für Crossbereich XT
 - Eclipse Rich Client Platform
 - JEE
 - Modellgetriebene Entwicklung
 - iOS Entwicklung

- We teach the basic concepts and patterns
- Almost everything is learned by practice: "learning by doing"
- 30% Theory
- 70% Practice

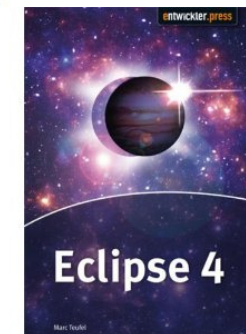
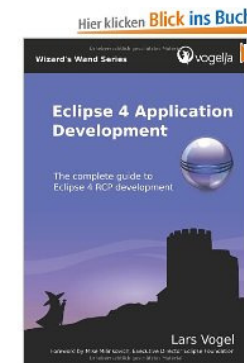
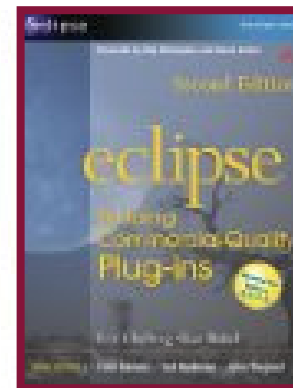
There are no stupid questions! (in English or in German)

XT ist der Cross-Bereich der msg zur

Sicherung der Wettbewerbsfähigkeit durch kontinuierliche
Verbesserung und Transfer von Software Engineering Kompetenz
für die Praxis



- Eclipse Rich Client Platform: Designing, Coding, and Packaging Java Applications
- **Eclipse. Building Commercial-Quality Plug-Ins (Eclipse (Addison-Wesley))**
- The Definitive Guide to SWT and JFace



Part I: Plugins and Features



- Eclipse IDE
- Structure of Eclipse
- Structure of a bundle
- Installing a plug-in

Structure of eclipse

- Install Eclipse with the PDE
- Have a look to the available views (shift+alt+q q)
- Install the demo plug-in
- Restart and check the views again

Ganymede Packages (based on Eclipse 3.4.1) - [Compare Packages](#)

[Lookin](#)



Eclipse IDE for Java EE Developers (163 MB)

Tools for Java developers creating JEE and Web applications, including a Java IDE, tools for JEE and JSF, Mylyn and others. [More...](#)

Downloads: 2,025,184

Windows
Mac OS X
Linux 32bit
Linux 64bit



Eclipse IDE for Java Developers (85 MB)

The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. [More...](#)

Downloads: 1,196,747

Windows
Mac OS X
Linux 32bit
Linux 64bit



Eclipse IDE for C/C++ Developers (68 MB)

An IDE for C/C++ developers with Mylyn integration. [More...](#)

Downloads: 626,276

Windows
Mac OS X
Linux 32bit
Linux 64bit



Featured Member Distro

Innovations

Download Eclipse IDE for Java EE Developers and Visual Rules. Business Rules Management with Eclipse. Fi trial.



Eclipse Modeling Tools (includes Incubating components) (300 MB)

This modeling package contains a collection of Eclipse Modeling Project components, including EMF, GMF, MDT XSD/OCL/UML2, M2M, M2T, and EMFT elements. It includes a complete SDK, developer tools and source code. Note that the Modeling package includes some *incubating* components, as indicated by feature numbers less than 1.0.0 on the feature list. [More...](#)

Downloads: 233,177

Windows
Mac OS X
Linux 32bit
Linux 64bit



Eclipse for RCP/Plug-in Developers (175 MB)

A complete set of tools for developers who want to create Eclipse plug-ins or Rich Client Applications. It includes a complete SDK, developer tools and source code, plus Mylyn, an XML editor and the Eclipse Communication Framework. [More...](#)

Downloads: 142,422

Windows
Mac OS X
Linux 32bit
Linux 64bit



Eclipse IDE for Java and Report Developers (195 MB)

JEE tools and BIRT reporting tool for Java developers to create JEE and Web applications that also have reporting needs. [More...](#)

Downloads: 111,167

Windows
Mac OS X
Linux 32bit
Linux 64bit



Eclipse Classic 3.4.1 (151 MB)

The classic Eclipse download: the Eclipse Platform, Java Development Tools, and Plug-in Development Environment, including source and both user and programmer documentation. [More...](#)

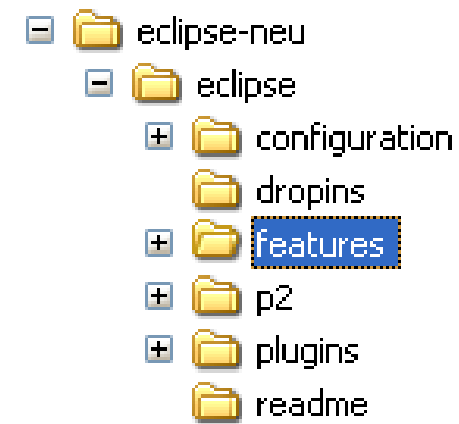
[Release notes](#) | [Other downloads](#) | [Documentation](#)

Downloads: 1,582,207

Windows
Mac OS X
Linux 32bit
Linux 64bit

Structure of eclipse

- What is a plug-in?
- What is a feature?

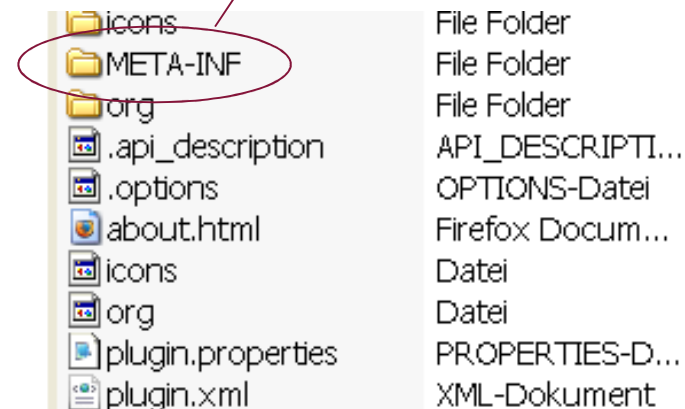
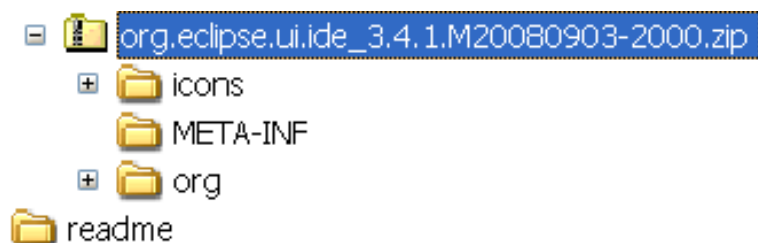


Plug-in structure

- org.eclipse.ui.intro.universal_3.2.200.v20080508
- org.eclipse.ui.workbench.compatibility_3.2.0.I20080509-2000
- org.junit4_4.3.1
- org.junit_3.8.2.v20080602-1318
- com.ibm.icu.source_3.8.1.v20080530.jar
- com.ibm.icu_3.8.1.v20080530.jar
- com.jcraft.jsch_0.1.37.v200803061811.jar
- javax.servlet.jsp.source_2.0.0.v200806031607.jar

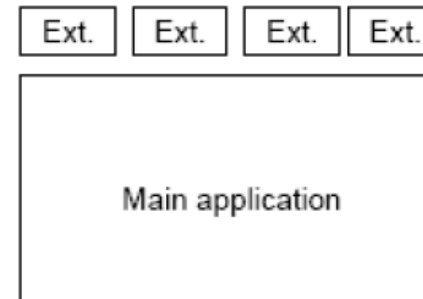
MANIFEST.MF

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Rcp Plug-in
Bundle-SymbolicName: my.first.rcp; singleton:=true
Bundle-Version: 1.0.0
Bundle-Activator: my.first.rcp.Activator
Require-Bundle: org.eclipse.ui,
org.eclipse.core.runtime
Eclipse-LazyStart: true
```

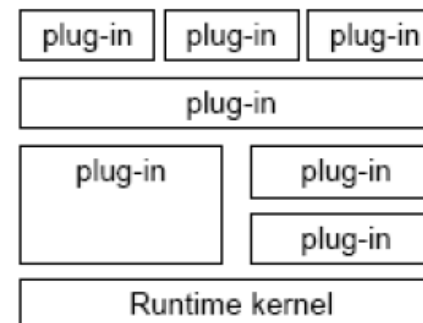


Extensible application vs. platform

- » Extensible applications (like AutoCAD and Photoshop) have add-on concepts.



- » The Eclipse Rich Client Platform is an open, extensible architecture built out of layers of plug-ins.
- » RCP is not a monolithic system!



Feature structure



General Information

This section describes general information about this feature.

ID:
Version:
Name:
Provider:
Branding Plug-in:
Update Site URL:
Update Site Name:

Supported Environments

Specify environment combinations in which this feature can be installed. Leave blank if the feature does not contain platform-specific code.

Operating Systems:
Window Systems:
Languages:
Architecture:

Information

Enter description, license and copyright information. Optionally, provide links to update sites for installing additional features.

Optional URL:

Text:

feature.xml
MANIFEST.MF

Plug-ins and Fragments

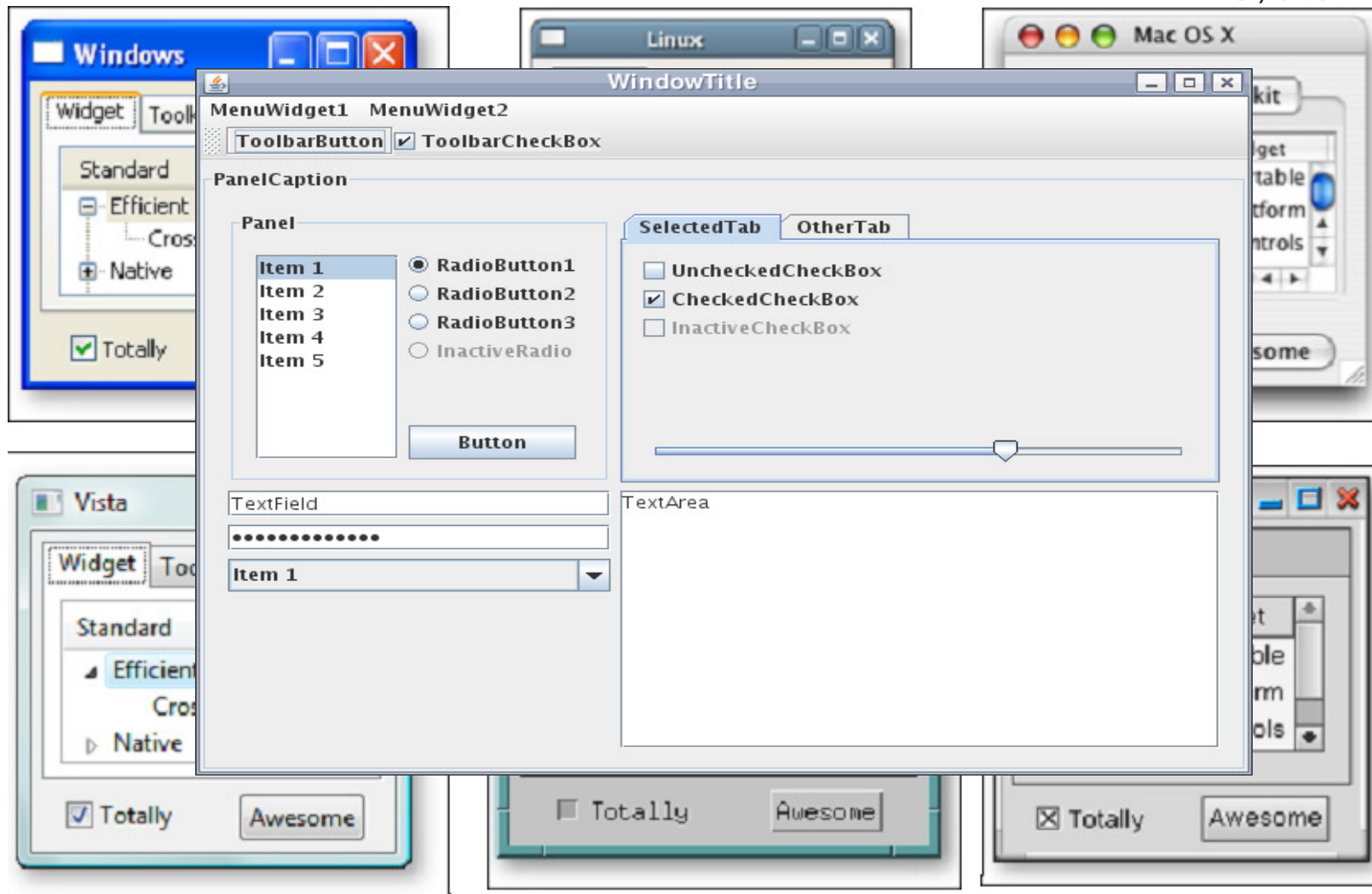
Select plug-ins and fragments that should be packaged in this feature.

What is a feature



- A feature is a grouping of plug-ins described in the feature manifest file (feature.xml)
- The manifest file is a list of plug-ins and the information for updating them
- Therefore features are the basis for using the Eclipse Update Manager
- It does not contain any code; code is kept in plug-ins

What is SWT/JFace/Eclipse Framework



Eclipse Forms

JFace

SWT

Package Explorer

- JRE System Library [JavaSE-1.6]
- Plug-in Dependencies
- src
- icons
- META-INF
- MANIFEST.MF
- build.properties
- plugin.xml
- com.msg.sample
- JRE System Library [JavaSE-1.6]
- Plug-in Dependencies
- src
 - com.msg.sample
 - Activator.java
 - com.msg.sample.views
 - SampleView.java
- icons
 - sample.gif
- META-INF
- MANIFEST.MF
- build.properties
- contexts.xml
- plugin.xml
- net.eclipse.training.pm
- net.eclipse.training.pm-feature
- net.eclipse.training.pm.core
- net.eclipse.training.pm.core.test
- net.eclipse.training.pm.helloWorld
- net.eclipse.training.pm.help
- net.eclipse.training.pm.listenerdemo
- net.eclipse.training.pm.releng
- StatusHandler
- JRE System Library [JavaSE-1.6]
- Plug-in Dependencies
- src
- icons
- META-INF
- MANIFEST.MF
- build.properties
- contexts.xml

Overview

General Information

This section describes general information about this plug-in.

ID: ttd

Version: 1.0.0

Name: Ttd Plug-in

Provider:

Platform Filter:

Activator: ttd.Activator [Browse...](#)

☒ Activate this plug-in when one of its classes is loaded

☒ This plug-in is a singleton

Execution Environments

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

Plug-in Content

The content of the plug-in is made up of two sections:

- [Dependencies](#): lists all the plug-ins required on this
- [Runtime](#): lists the libraries that make up this plug-in

Extension / Extension Point Content

This plug-in may define extensions and extension point

- [Extensions](#): declares contributions this plug-in makes
- [Extension Points](#): declares new function points that can be extended

Testing

Test this plug-in by launching a separate Eclipse application

- [Launch an Eclipse application](#)
- [Launch an Eclipse application in Debug mode](#)

Search

File Search Task Search Java Search Plug-in Search

Containing text:

☐ Case sensitive

☐ Regular expression

(* = any string, ? = any character, \ = escape for literals: * ? \)

File name patterns:

[Choose...](#)

Patterns are separated by a comma (* = any string, ? = any character)

☐ Consider derived resources

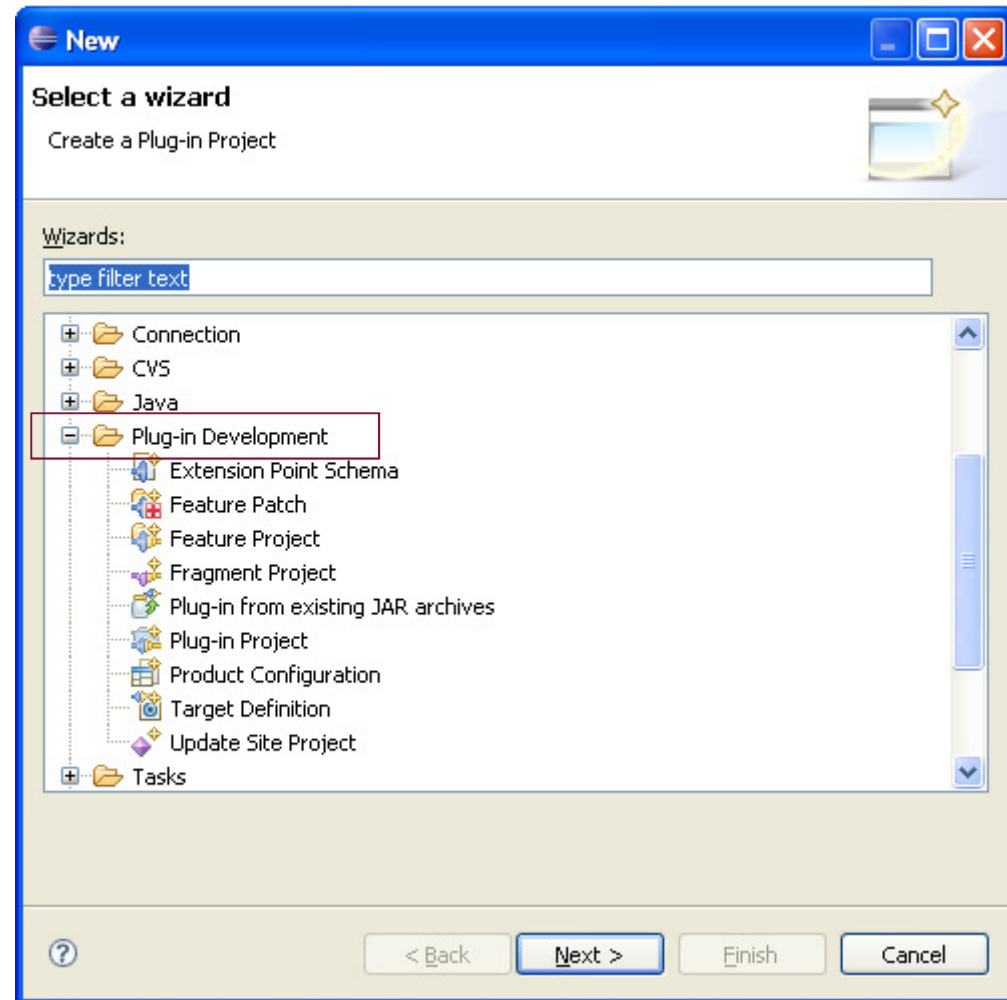
Scope

☒ Workspace ☐ Selected resources ☐ Enclosing projects

☐ Working set: [Choose...](#)

Location	Type
ainin... line 28	Java Task
ainin... line 33	Java Task
ainin... line 31	Java Task
ainin... line 76	Java Task
ainin... line 24	Java Task
ainin... line 28	Java Task
ainin... line 57	Java Task

Let's start with our first plug-in



New Plug-in Project

Plug-in Project

Create a new plug-in project

Project name:

☒ Use default location

Location:

Project Settings

☒ Create a Java project

Source folder:

Output folder:

Target Platform

This plug-in is targeted to run with:

☒ Eclipse version:

☐ an OSGi framework:

Working sets

☐ Add project to working sets

Working sets:

New Plug-in Project

Plug-in Content

Enter the data required to generate the plug-in.

Plug-in Properties

Plug-in ID:

Plug-in Version:

Plug-in Name:

Plug-in Provider:

Execution Environment:

Plug-in Options

☒ Generate an activator, a Java class that controls the plug-in's life cycle

Activator:

☒ This plug-in will make contributions to the UI

☐ Enable API Analysis

Rich Client Application

Would you like to create a rich client application? ☐ Yes ☒ No

New Plug-in Project

Templates

Select one of the available templates to generate a fully-functioning plug-in.

☒ Create a plug-in using one of the templates

Available Templates:

- Custom plug-in wizard
- Hello, World
- Hello, World Command
- Plug-in with a multi-page editor
- Plug-in with a popup menu
- Plug-in with a property page
- Plug-in with a view
- Plug-in with an editor
- Plug-in with an incremental project builder
- Plug-in with sample help content

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

- Sample view.** This template creates a workbench view. The view is contributed to the workbench by creating a category. The view can be opened by selecting **Window, Show View** and then **Other...** on the menu bar. The template demonstrates implementation of pop-up menu support, local tool bar, double-click, sorting and filtering. There is also an option to add context-sensitive help to the view.

Extensions Used

- org.eclipse.ui.views
- org.eclipse.ui.perspectiveExtensions
- org.eclipse.help.contexts

New plug-in project with a sample view

Main View Settings

Choose the way the new view will be added to the plug-in.

Java Package Name:

View Class Name:

View Name:

View Category ID:

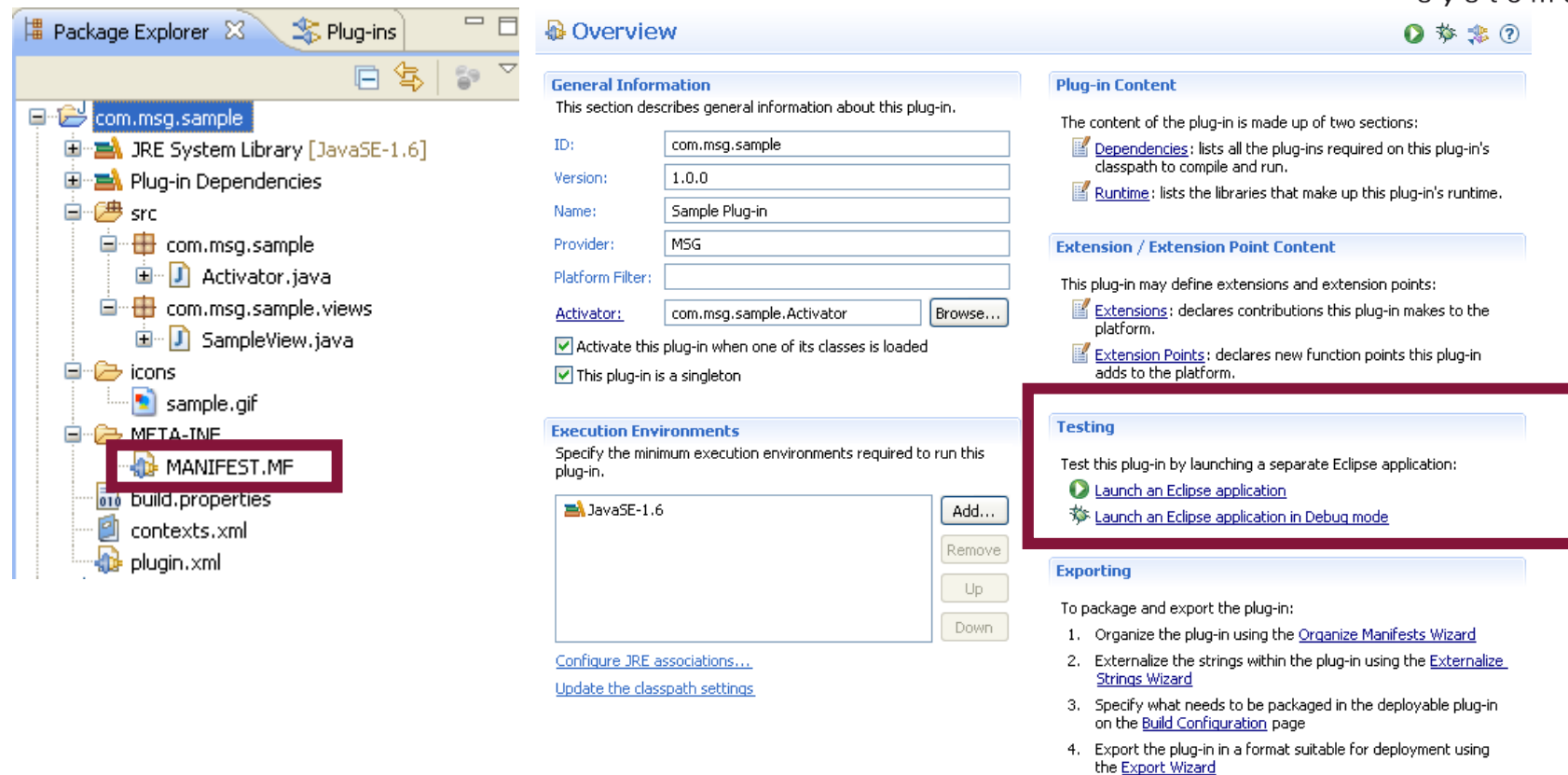
View Category Name:

Select the viewer type that should be hosted in the view:

☒ Table viewer (can also be used for lists) ☐ Tree viewer

☒ Add the view to the java perspective

☒ Add context help to the view



Package Explorer

- com.msg.sample
 - JRE System Library [JavaSE-1.6]
 - Plug-in Dependencies
 - src
 - com.msg.sample
 - Activator.java
 - com.msg.sample.views
 - SampleView.java
 - icons
 - sample.gif
 - META-INF
 - MANIFEST.MF**
 - build.properties
 - contexts.xml
 - plugin.xml

Overview

General Information
This section describes general information about this plug-in.

ID: com.msg.sample
Version: 1.0.0
Name: Sample Plug-in
Provider: MSG
Platform Filter:
Activator: com.msg.sample.Activator [Browse...](#)
☒ Activate this plug-in when one of its classes is loaded
☒ This plug-in is a singleton

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

JavaSE-1.6 [Add...](#) [Remove](#) [Up](#) [Down](#)
[Configure JRE associations...](#)
[Update the classpath settings](#)

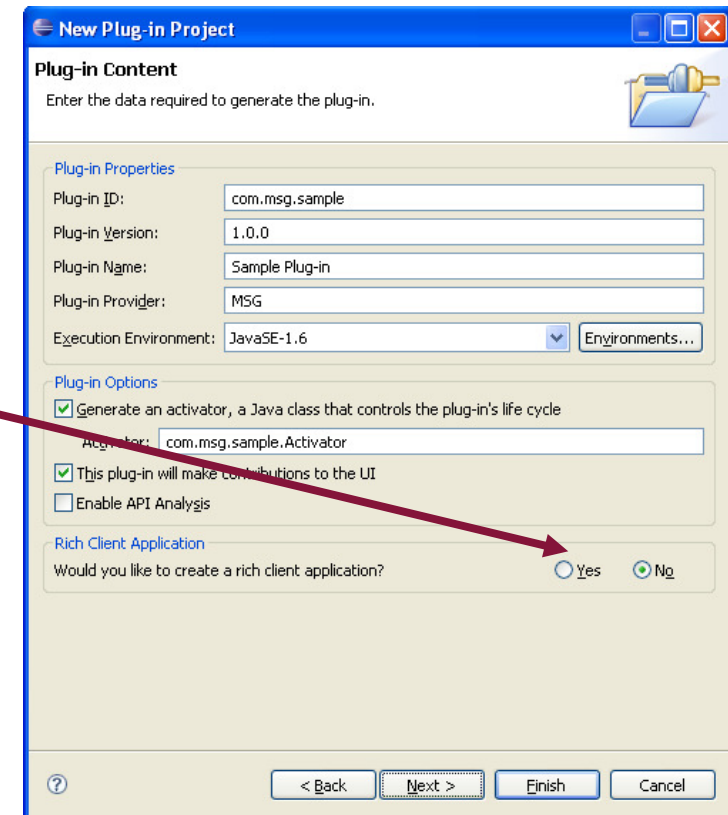
Plug-in Content
The content of the plug-in is made up of two sections:
[Dependencies](#): lists all the plug-ins required on this plug-in's classpath to compile and run.
[Runtime](#): lists the libraries that make up this plug-in's runtime.

Extension / Extension Point Content
This plug-in may define extensions and extension points:
[Extensions](#): declares contributions this plug-in makes to the platform.
[Extension Points](#): declares new function points this plug-in adds to the platform.

Testing
Test this plug-in by launching a separate Eclipse application:
[Launch an Eclipse application](#)
[Launch an Eclipse application in Debug mode](#)

Exporting
To package and export the plug-in:
1. Organize the plug-in using the [Organize Manifests Wizard](#)
2. Externalize the strings within the plug-in using the [Externalize Strings Wizard](#)
3. Specify what needs to be packaged in the deployable plug-in on the [Build Configuration](#) page
4. Export the plug-in in a format suitable for deployment using the [Export Wizard](#)

- Create your first plug-in
- Launch the IDE and try to open your view
- Create another plug-in but
Check the “rich client application”
Checkbox
- Launch the application
- Create a plug-in using the template
“Hello, world command”
and test it



Vielen Dank für Ihre Aufmerksamkeit



.consulting .solutions .partnership

