



编写、导出及安装Eclipse插件

软件所智能软件中心PLCT实验室 张爱珩 实习生

目录

01 创建Plugins项目

02 编写插件

03 运行调试插件

04 导出、使用插件

01 创建Plugins项目

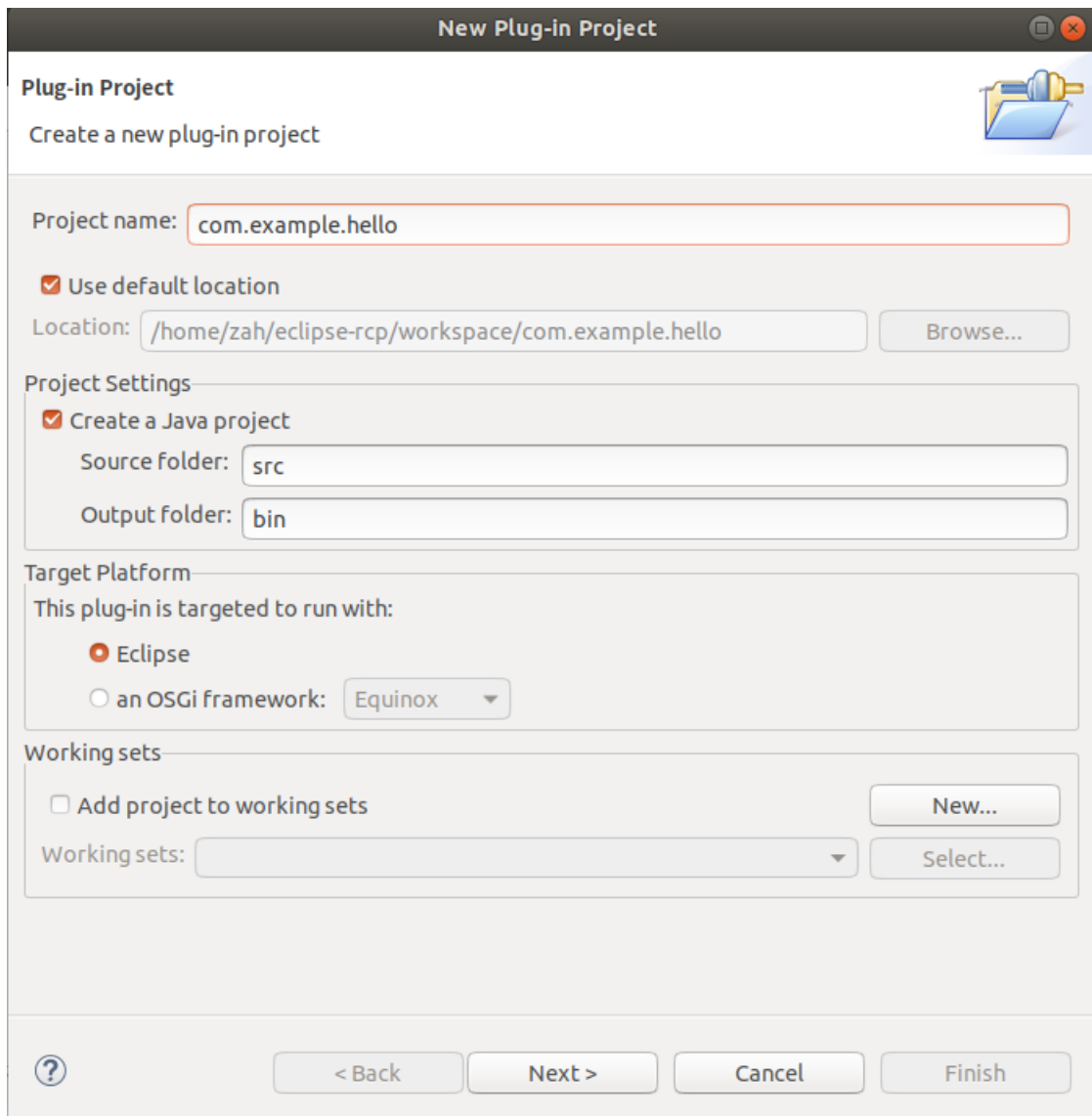
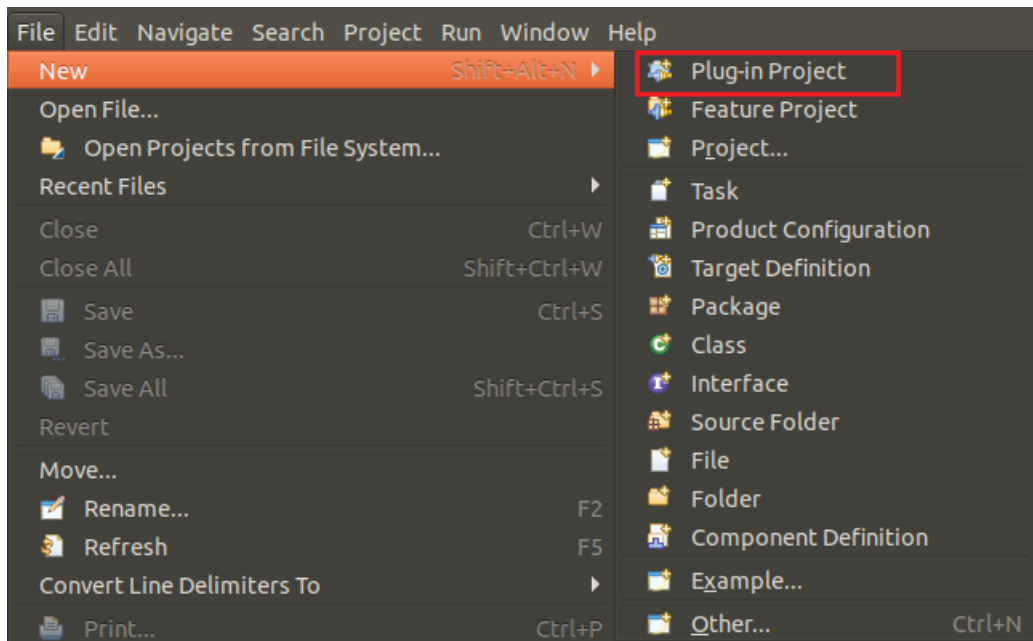
安装开发eclipse插件所需IDE

- 下载地址: <http://mirrors.ustc.edu.cn/eclipse/technology/epp/downloads/release/2019-12/R/>
- Eclipse RCP 版本

| | | |
|---|-------------------|-----------|
| eclipse-rcp-2019-12-R-linux-gtk-x86_64.tar.gz | 13-Dec-2019 12:00 | 318131838 |
| eclipse-rcp-2019-12-R-linux-gtk-x86_64.tar.gz.md5 | 13-Dec-2019 09:38 | 80 |
| eclipse-rcp-2019-12-R-linux-gtk-x86_64.tar.gz.sha1 | 13-Dec-2019 09:38 | 88 |
| eclipse-rcp-2019-12-R-linux-gtk-x86_64.tar.gz.s..> | 13-Dec-2019 09:38 | 176 |
| eclipse-rcp-2019-12-R-macosx-cocoa-x86_64.dmg | 13-Dec-2019 12:00 | 319286726 |
| eclipse-rcp-2019-12-R-macosx-cocoa-x86_64.dmg.md5 | 13-Dec-2019 12:47 | 80 |
| eclipse-rcp-2019-12-R-macosx-cocoa-x86_64.dmg.sha1 | 13-Dec-2019 12:47 | 88 |
| eclipse-rcp-2019-12-R-macosx-cocoa-x86_64.dmg.s..> | 13-Dec-2019 12:47 | 176 |
| eclipse-rcp-2019-12-R-win32-x86_64.zip | 13-Dec-2019 12:00 | 320196210 |
| eclipse-rcp-2019-12-R-win32-x86_64.zip.md5 | 13-Dec-2019 09:37 | 73 |
| eclipse-rcp-2019-12-R-win32-x86_64.zip.sha1 | 13-Dec-2019 09:37 | 81 |
| eclipse-rcp-2019-12-R-win32-x86_64.zip.sha512 | 13-Dec-2019 09:37 | 169 |

01 创建Plugins项目

创建Plugins开发项目



01 创建Plugins项目

创建Plugins开发项目

New Plug-in Project

Content
Enter the data required to generate the plug-in.

Properties

ID:

Version:

Name:

Vendor:

Execution Environment:

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 Plug-in Templates:

- Custom plug-in wizard
- Editor contribution for XML files
- Hello, World Command**
- Incremental project builder
- Menu contribution using 4.x API
- Multi-page editor
- Property page
- RAP e4 Template
- RAP Hello World
- RAP Mail Template
- Sample help content
- Textual editor, relying on Generic Editor
- Toolbar contribution using e4 API
- View contribution using 3.x API
- View contribution using 4.x API

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

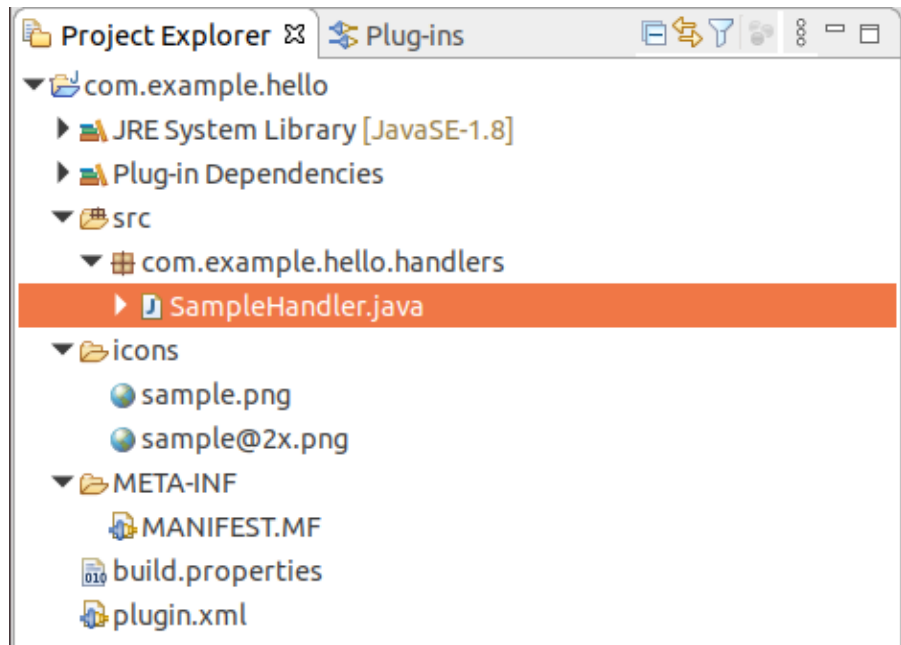
- Command contribution.** This template creates a simple command contribution that adds **Sample Menu** to the menu bar and a button to the toolbar. Both the menu item in the new menu and the button invoke the same **Sample Action**. Its role is to open a simple message dialog with a message of your choice.

Extensions Used

- org.eclipse.ui.commands
- org.eclipse.ui.handlers
- org.eclipse.ui.bindings
- org.eclipse.ui.menus

项目目录结构

- src : 项目源代码目录
- icons : 项目图片资源目录
- META-INF/MANIFEST.MF : 项目基本配置信息, 版本、名称、启动器等
- build.properties : 项目的编译配置信息, 包括, 源代码路径、输出路径等
- plugin.xml : 插件的操作配置信息, 包含弹出菜单及点击菜单后对应的操作执行类等
- plugin manifest editor



参考资料:

- [1] <https://www.cnblogs.com/haifeng1990/p/5192804.html>
- [2] <https://www.cnblogs.com/lotuses/p/11561457.html>
- [3] <https://www.ibm.com/developerworks/cn/java/os-ecplug/>

MANIFEST.MF

com.example.hello SampleHandler.java

Overview

General Information

This section describes general information about this plug-in.

ID:

Version:

Name:

Vendor:

Platform Filter:

Activator:

☐ 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-1.8

[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 a RAP Application](#)

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

[Launch a RAP Application in Debug mode](#)

Exporting

Overview

Dependencies

Runtime

Extensions

Extension Points

Build

MANIFEST.MF

plugin.xml

build.properties

Dependencies

Required Plug-ins

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

org.eclipse.ui

```
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Name: Hello
4 Bundle-SymbolicName: com.example.hello;singleton:=true
5 Bundle-Version: 1.0.0.qualifier
6 Bundle-Vendor: EXAMPLE
7 Require-Bundle: org.eclipse.ui
8 Automatic-Module-Name: com.example.hello
9 Bundle-RequiredExecutionEnvironment: JavaSE-1.8
10
```

build.propertyis

Build Configuration

Custom Build

Runtime Information

Define the libraries, specify the order in which they should be built, and list the source folders that should be compiled into each selected library.

Add Library...

Up

Down

src/

Add Folder...

Binary Build

Select the folders and files to include in the binary build.

☐ .classpath

☐ .project

.settings

☒ META-INF

bin

☐ build.properties

☒ icons

☒ plugin.xml

☐ src

Source Build

Select the folders and files to include in the source build.

☐ .classpath

☐ .project

.settings

☐ META-INF

bin

☐ build.properties

☐ icons

☐ plugin.xml

☐ src

Extra Classpath Entries

Overview

Dependencies

Runtime

Extensions

Extension Points

Build

MANIFEST.MF

plugin.xml

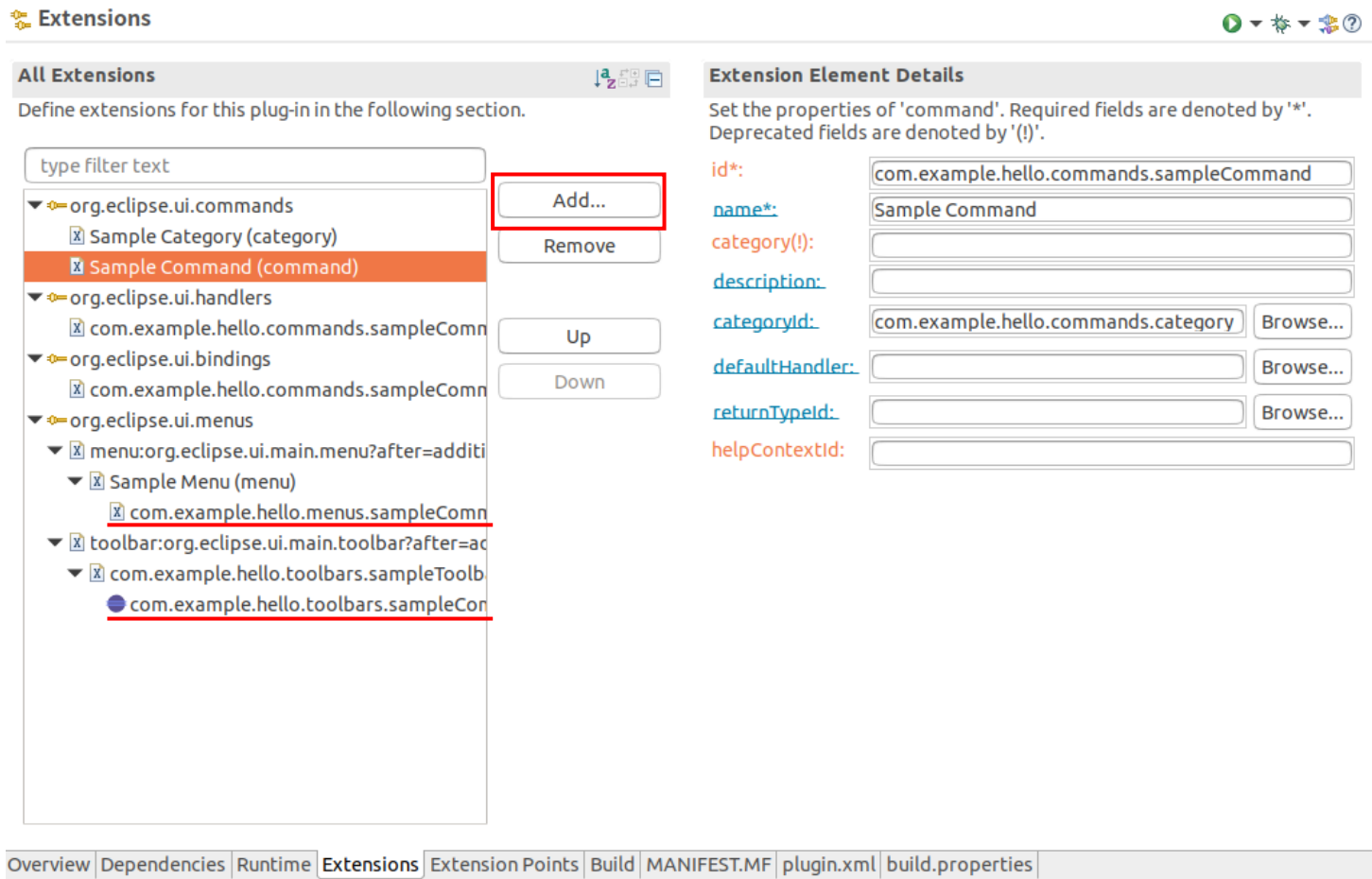
build.properties

```
1 source.. = src/
2 output.. = bin/
3 bin.includes = plugin.xml,\
4               META-INF/,\
5               .,\
6               icons/
7
```


Extensions

创建菜单所需的三个扩展点 (Extensions)

- Menus : 添加菜单
- Commands : 声明一个行为操作
- Handlers : 具体行为操作实现

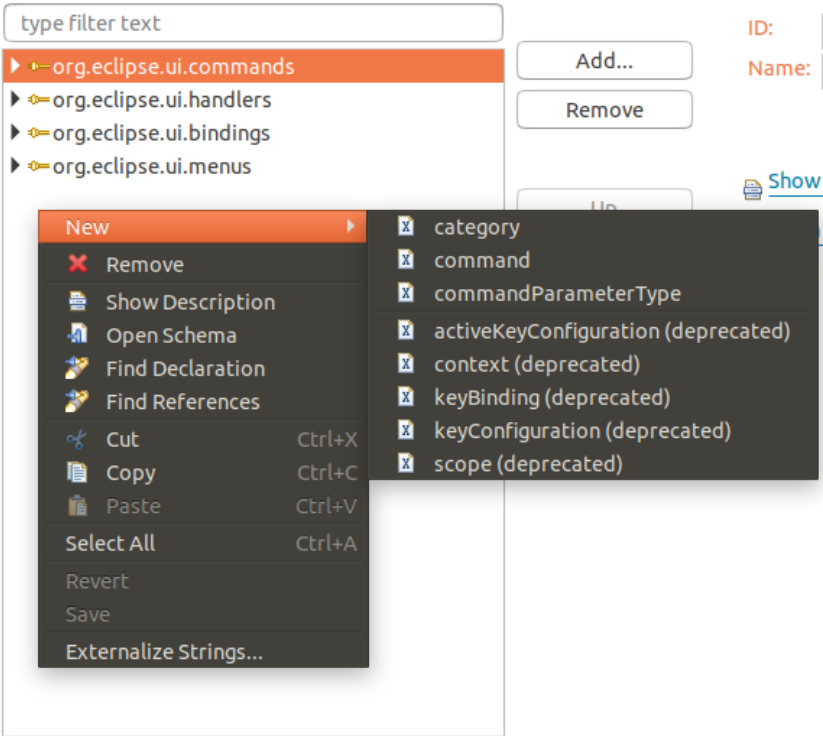
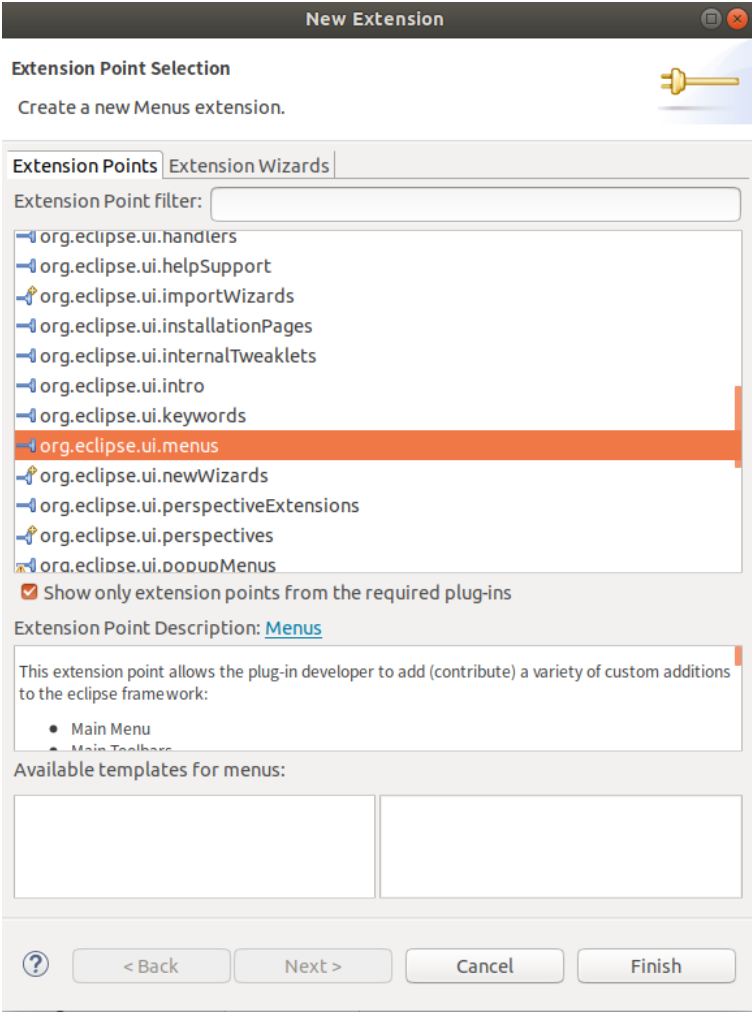


参考资料:

[1] <https://www.cnblogs.com/haifeng1990/p/5198449.html>

[2] <https://www.cnblogs.com/lotuses/p/11561457.html>

Extensions



02 编写插件

plugin.xml

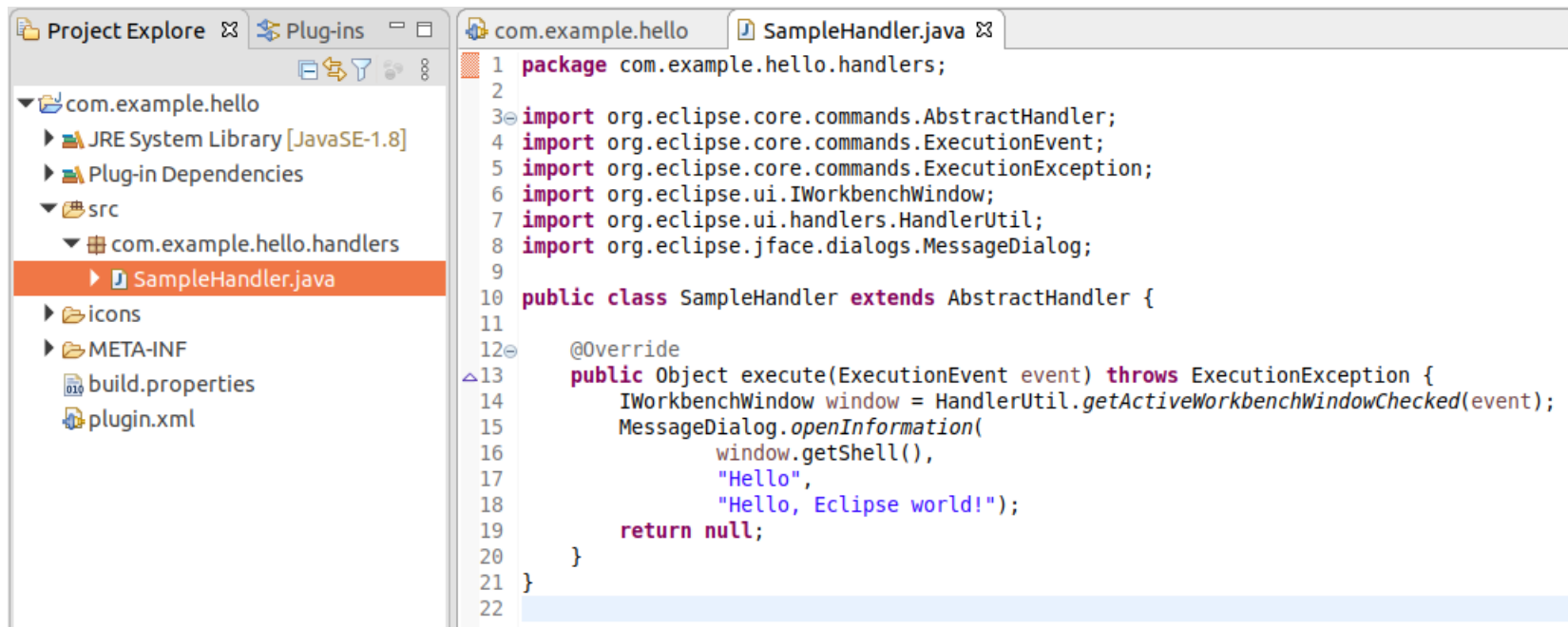
```
com.example.hello  SampleHandler.java
1<?xml version="1.0" encoding="UTF-8"?>
2<?eclipse version="3.4"?>
3<plugin>
4
5  <extension
6    point="org.eclipse.ui.commands">
7    <category
8      id="com.example.hello.commands.category"
9      name="Sample Category">
10   </category>
11   <command
12     categoryId="com.example.hello.commands.category"
13     name="Sample Command"
14     id="com.example.hello.commands.sampleCommand">
15   </command>
16 </extension>
17 <extension
18   point="org.eclipse.ui.handlers">
19   <handler
20     class="com.example.hello.handlers.SampleHandler"
21     commandId="com.example.hello.commands.sampleCommand">
22   </handler>
23 </extension>
24 <extension
25   point="org.eclipse.ui.bindings">
26   <key
27     commandId="com.example.hello.commands.sampleCommand"
28     schemeId="org.eclipse.ui.defaultAcceleratorConfiguration"
29     contextId="org.eclipse.ui.contexts.window"
30     sequence="M1+6">
31   </key>
32 </extension>
33 <extension
34   point="org.eclipse.ui.menus">
35   <menuContribution
36     locationURI="menu:org.eclipse.ui.main.menu?after=additions">
37   <menu
38     id="com.example.hello.menus.sampleMenu"
```

Overview Dependencies Runtime Extensions Extension Points Build MANIFEST.MF plugin.xml build.properties

02 编写插件

SampleHandler.java

- AbstractHandler：这个抽象实现为处理程序监听器提供了支持。
- ExecutionEvent：传递数据对象及其处理程序。
- ExecutionException：指示在执行命令期间发生异常。
- IWorkbenchWindow：工作台中的顶层窗口的实现类。
- HandlerUtil：在图形化界面中使用处理程序的一些常见实用程序。
- MessageDialog：向用户显示消息的对话框类。

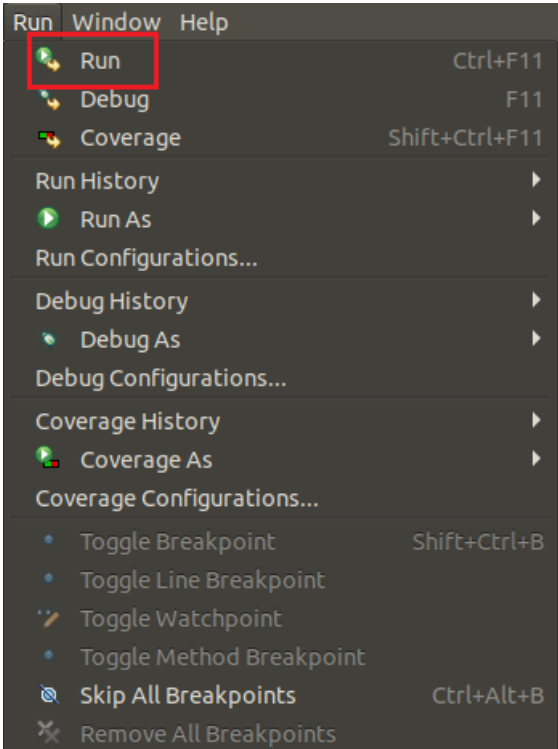
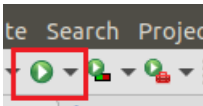


The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure for 'com.example.hello'. The 'src' folder is expanded, showing the 'com.example.hello.handlers' package, which contains the 'SampleHandler.java' file. On the right, the editor window shows the code for 'SampleHandler.java'. The code defines a package, imports several Eclipse API classes, and implements the 'execute' method of the 'AbstractHandler' interface.

```
1 package com.example.hello.handlers;
2
3 import org.eclipse.core.commands.AbstractHandler;
4 import org.eclipse.core.commands.ExecutionEvent;
5 import org.eclipse.core.commands.ExecutionException;
6 import org.eclipse.ui.IWorkbenchWindow;
7 import org.eclipse.ui.handlers.HandlerUtil;
8 import org.eclipse.jface.dialogs.MessageDialog;
9
10 public class SampleHandler extends AbstractHandler {
11
12     @Override
13     public Object execute(ExecutionEvent event) throws ExecutionException {
14         IWorkbenchWindow window = HandlerUtil.getActiveWorkbenchWindowChecked(event);
15         MessageDialog.openInformation(
16             window.getShell(),
17             "Hello",
18             "Hello, Eclipse world!");
19         return null;
20     }
21 }
22
```

03 运行调试插件

运行插件



Overview

General Information

This section describes general information about this plug-in.

ID:

Version:

Name:

Vendor:

Platform Filter:

Activator:

☐ 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-1.8

[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 a RAP Application](#)

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

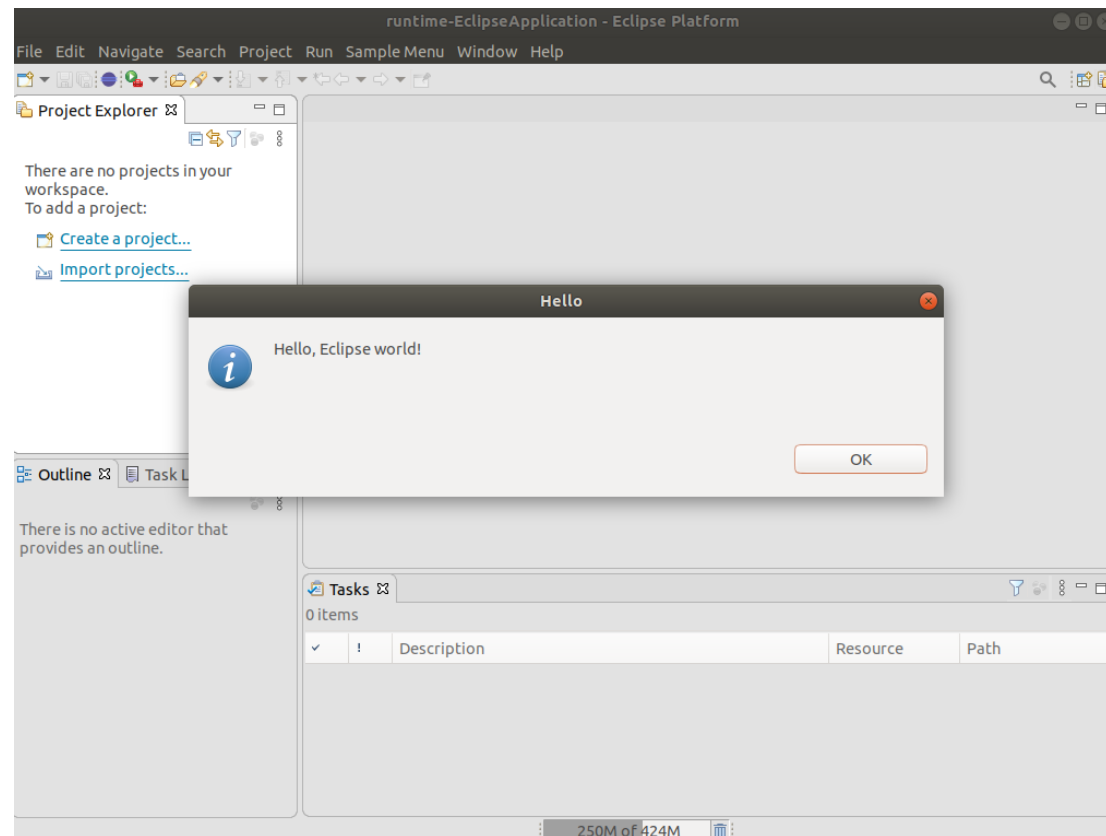
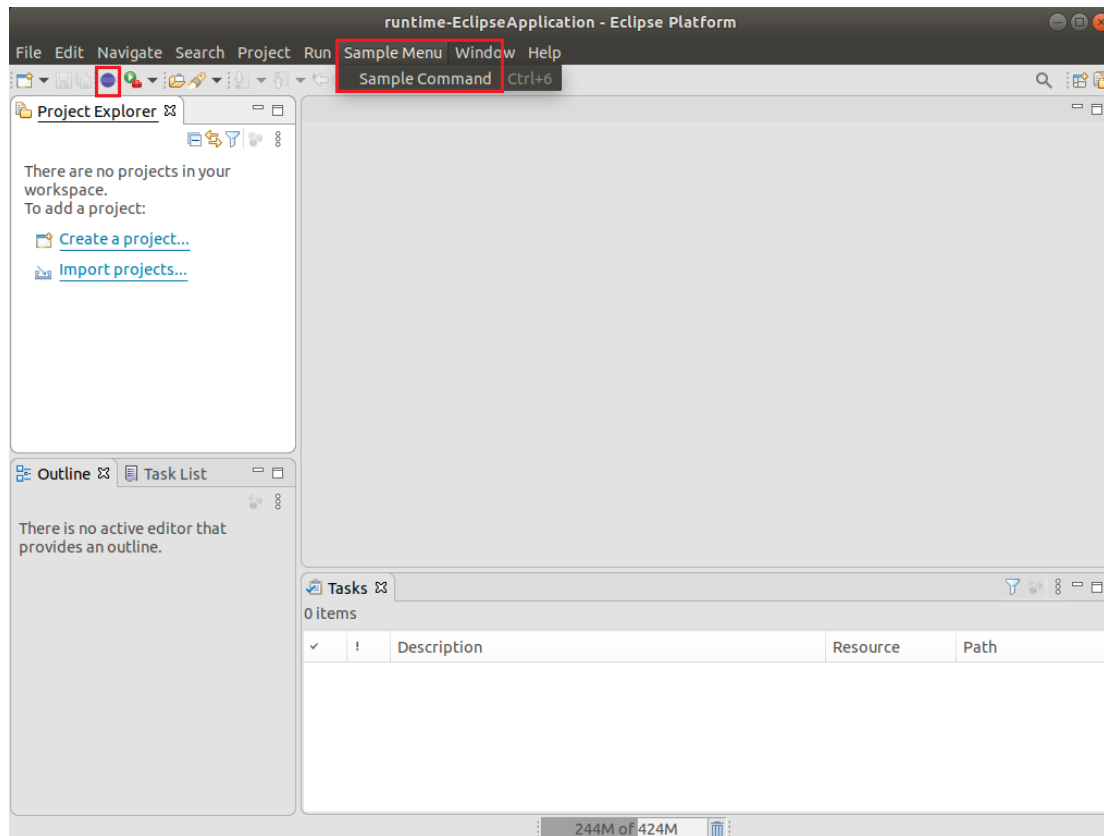
[Launch a RAP Application in Debug mode](#)

Exporting

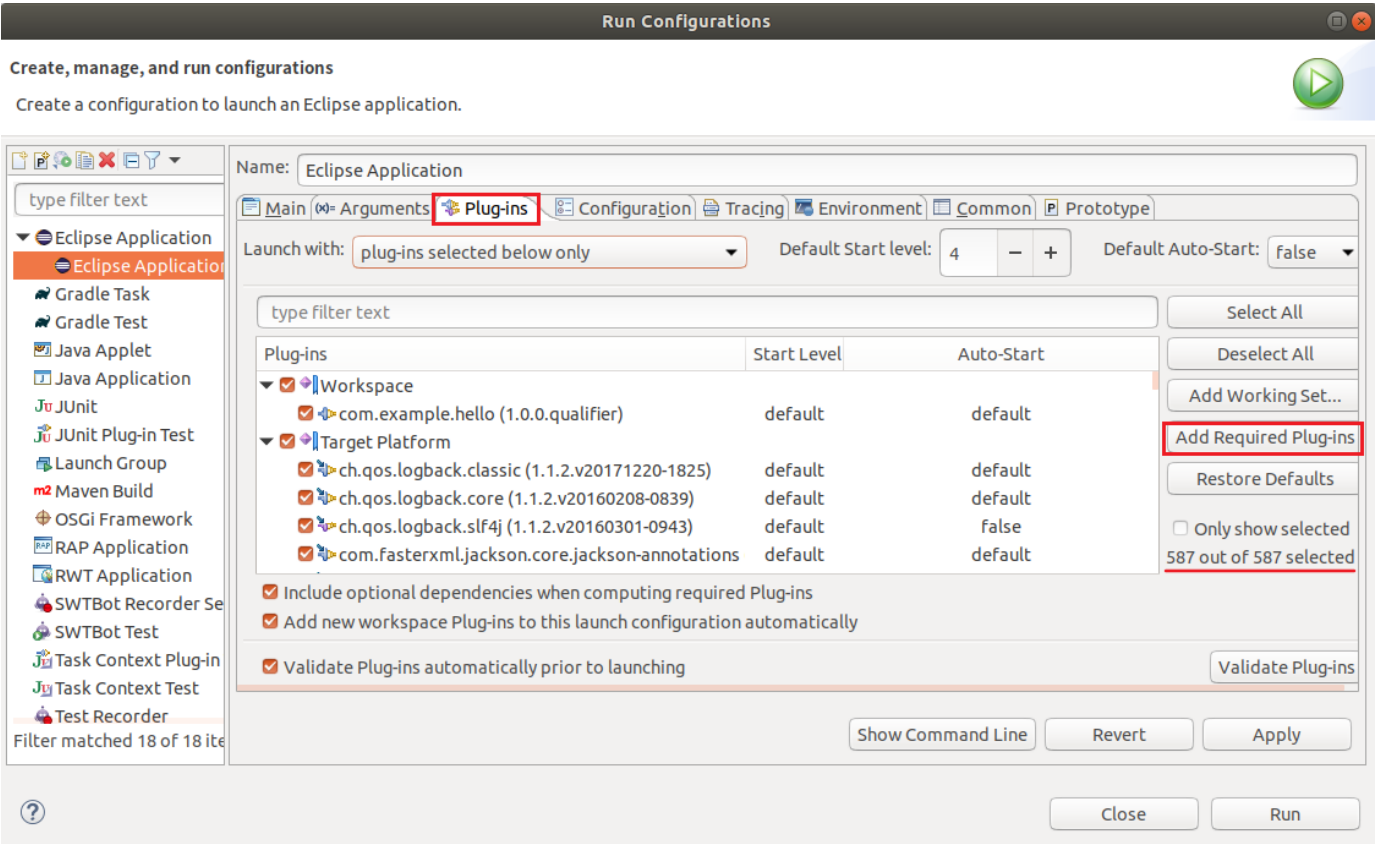
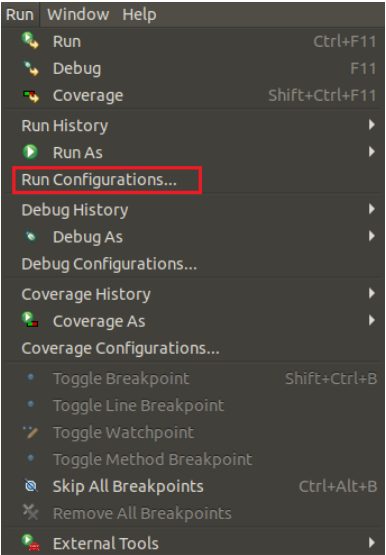
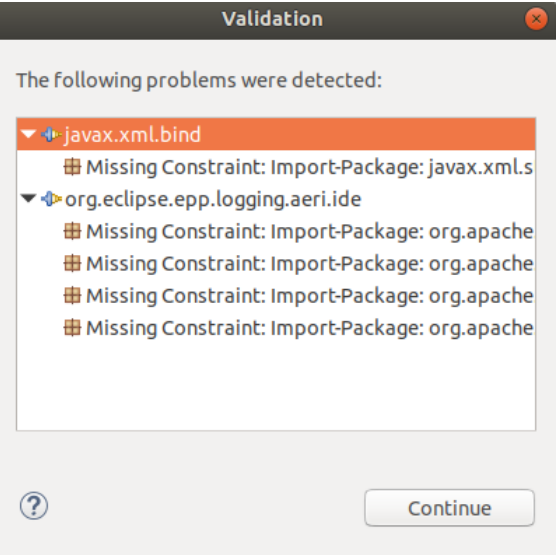
Overview | Dependencies | Runtime | Extensions | Extension Points | Build | MANIFEST.MF | plugin.xml | build.properties

03 运行调试插件

测试结果



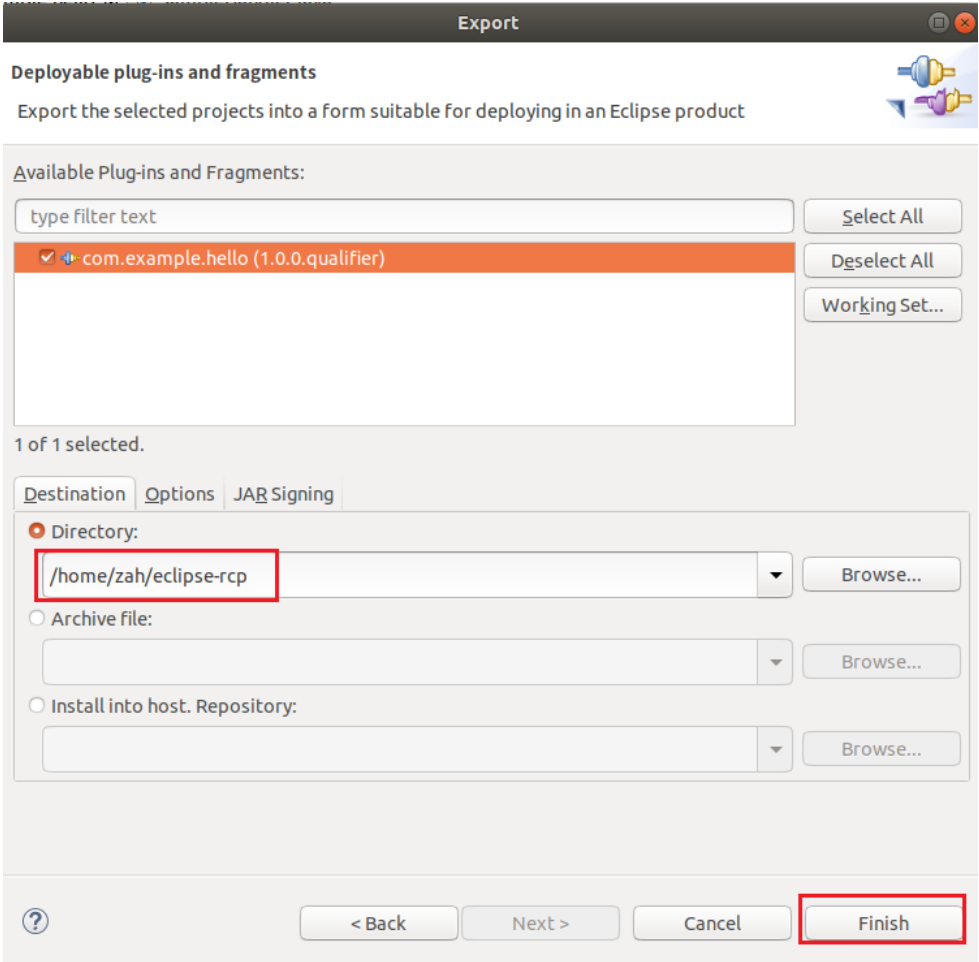
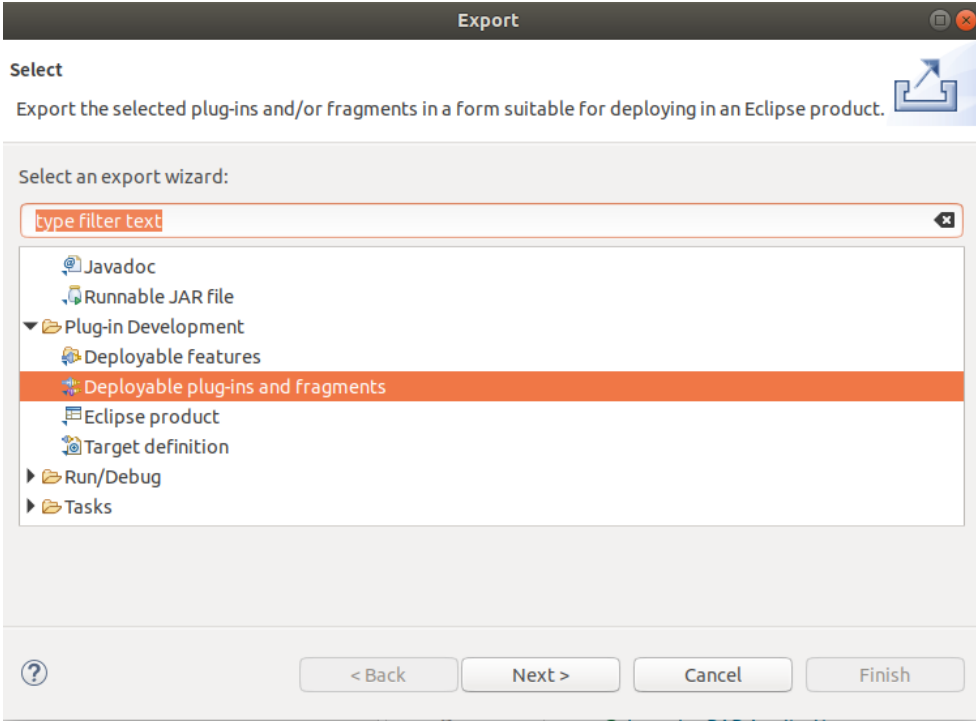
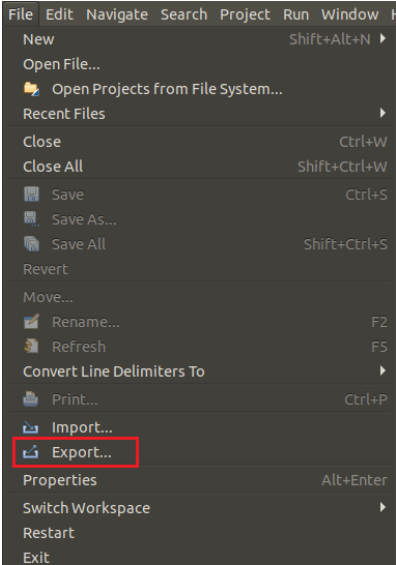
遇到的问题



参考资料：
[1] <https://blog.csdn.net/xiezhongweiwei/article/details/45465641>

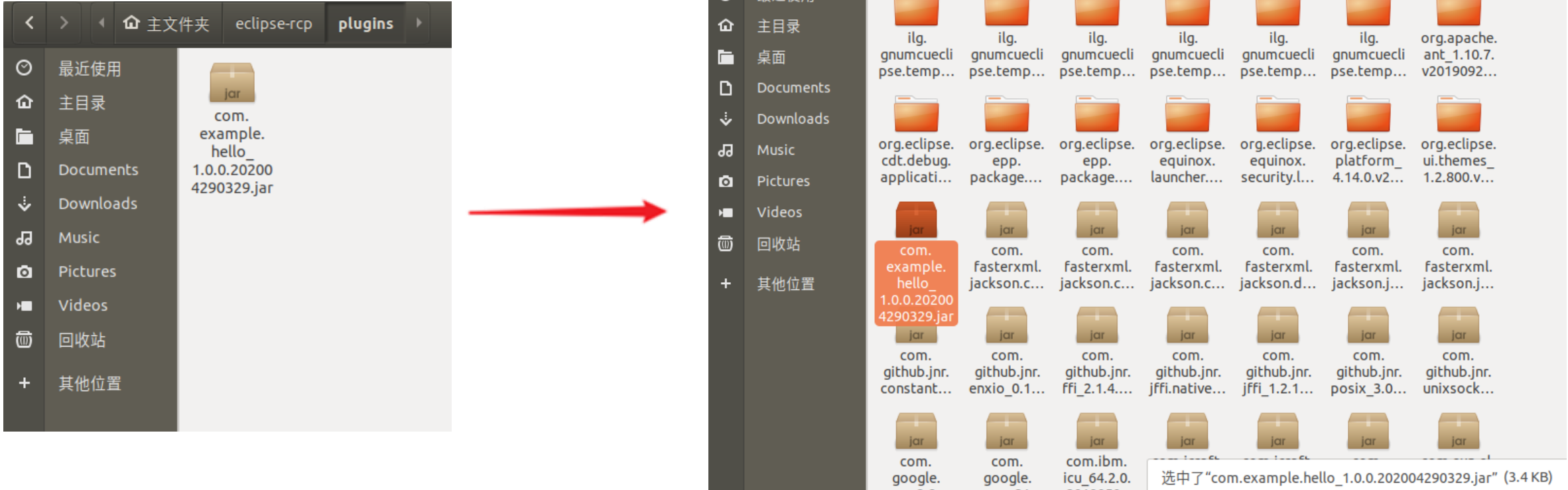
04 导出、使用插件

导出插件



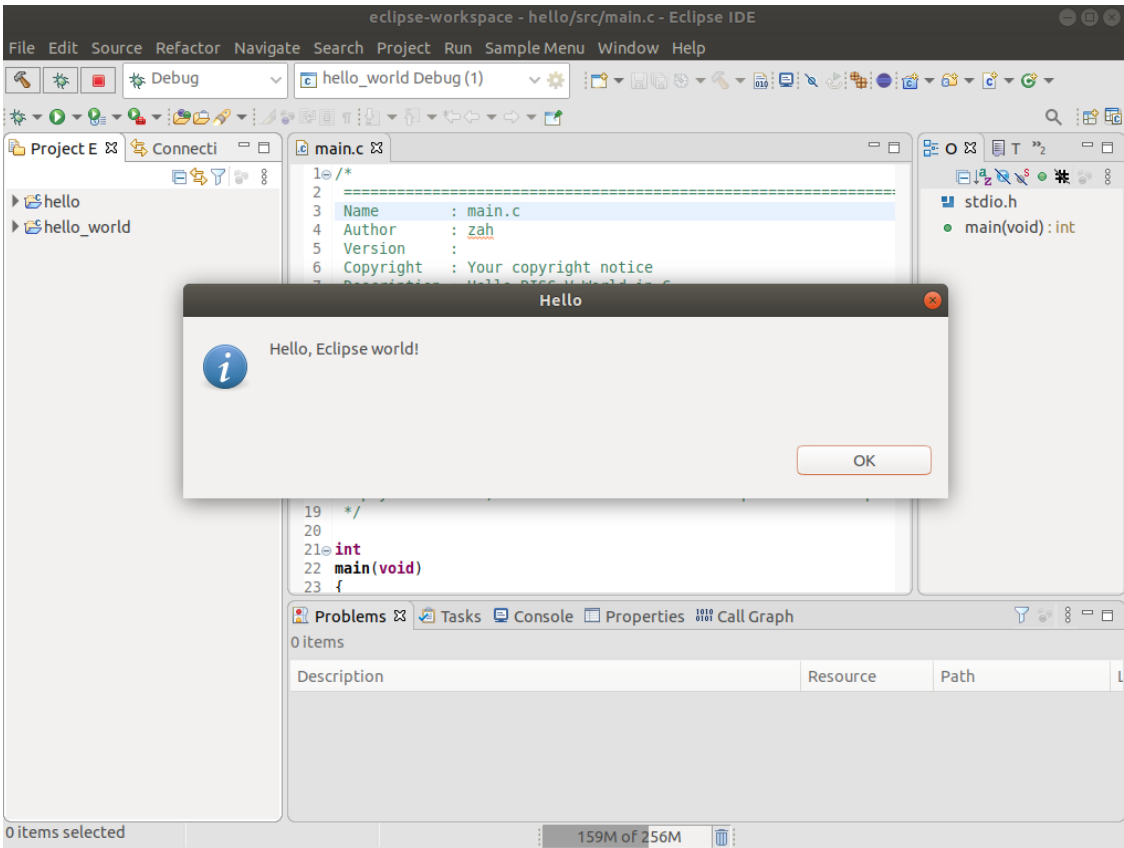
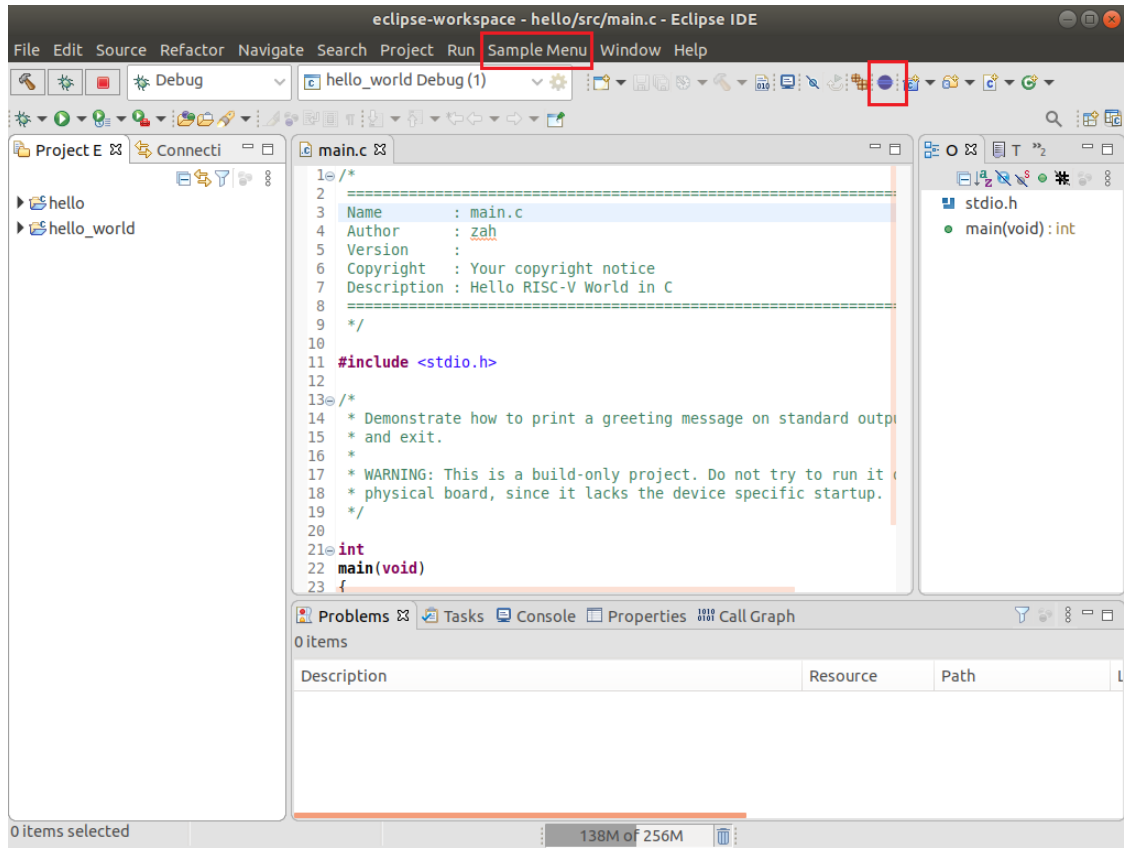
04 导出、使用插件

添加插件



04 导出、使用插件

使用插件



谢 谢

欢迎交流合作

2020/04/29