

Administration Extension Example	
AEE - Introduction	1
AEE - Java API Workspace preparation	1
AEE - Creating project plugin	1
AEE - Deployment to Installed Polarion	1
AEE - Execution from Workspace	1
AEE - Configuration	1

See also main SDK page.

AEE - Introduction

In this example we will show how to create custom administration page.

what will be shown in the example

1. how to create custom administration page
2. how to register custom administration page to Polarion

AEE - Java API Workspace preparation

See section *Workspace preparation*

AEE - Creating project plugin

The best way is to import provided example, which contains all dependencies you need for custom plugin creation

AEE - Import of the example

Info: You must ensure that your plugin is compiled against your Polarion version. This example contains precompiled jar plugin. You can remove it before you start developing your plugin based on this example. The Eclipse ensure that new jar plugin will be created against your source code and Polarion version.

To import this example to workspace, do these steps:

1. Select **File > Import...**
2. In the dialog that appears, select **Existing Project into Workspace** in **General** section and press **Next** button.
3. By pressing **Browse..** button, select the directory of examples (mostly in `C:\Polarion\polarion\SDK\examples\`). Submit it.
4. Select `com.polarion.example.administration` and press **Finish**.

AEE - Deployment to Installed Polarion

See section *Deployment to Installed Polarion*

AEE - Execution from Workspace

See section *Execution from Workspace*

AEE - Configuration

Custom administration pages are registered via `hivemodule.xml` to contribution with id `com.polarion.xray.webui.administrationPageExtenders`.

Extender attributes: id (required), name, iconUrl, pageUrl, parentNodeId, parentNodeName, parentNodeIconUrl, projectScope (Boolean), projectGroupScope (Boolean), repositoryScope (Boolean)

```
<?xml version="1.0" encoding="UTF-8"?>
<module id="com.polarion.example.administration" version="1.0.0">
  <contribution configuration-id="com.polarion.xray.webui.administrationPageExtenders">
    <extender id="administrationExample"
      parentNodeId="administrationExampleFolder"
      name="Page" parentNodeName="Example"
      parentNodeIconUrl="/polarion/icons/default/topicIcons/Tools_157-wrench.png"
      iconUrl="/polarion/icons/default/topicIconsSmall/Tools_158-wrench-2.png"
      pageUrl="/polarion/administrationExample/administration.jsp?scope=$scope$"
      projectScope="true" projectGroupScope="true" repositoryScope="true"/>
  </contribution>
</module>
```

Requirements

Development Environments

- [Eclipse IDE for Java EE Developers](#) or any other Eclipse IDE with The Eclipse Plug-in Development Environment (Go to Help > Install New Software... > Install Eclipse Plug-in Development Environment > Restart Eclipse)

- [Java Platform, Standard Edition 8 Development Kit](#) or Java bundled with your Polaron installation

Polarion Server

- Polaron Server with version 3.1.0 and higher

Workspace Preparation

To start developing a Polaron Java API plug-in, you first need to perform following steps:

1. Start Eclipse, then select **Window > Preferences...**
2. In the dialog that appears, select **Plug-In Development > Target Platform**.
3. Click the **Add** button on the right.
4. Keep the **Nothing: Start with an empty target definition** option selected and click **Next**.

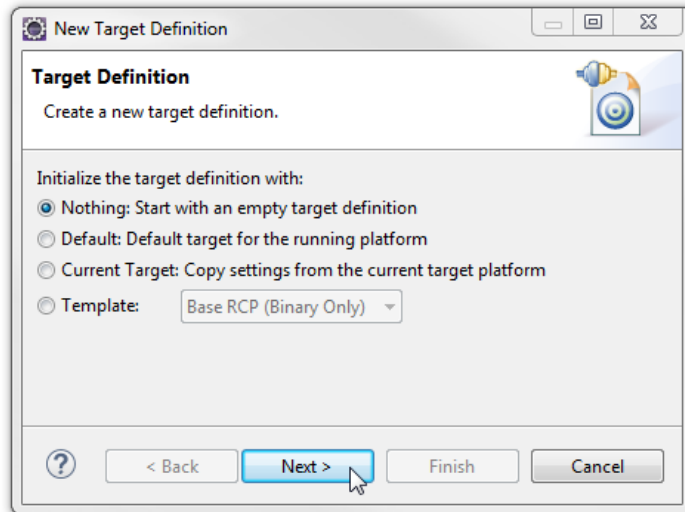



Figure WP-1: Starting with an Empty Target Definition

5. Enter a **Name** and click **Add**.
6. Select  **Directory** and click **Next**.
7. Click **Browse**, select the `C:\Polarion\polarion` folder and click **Next**.

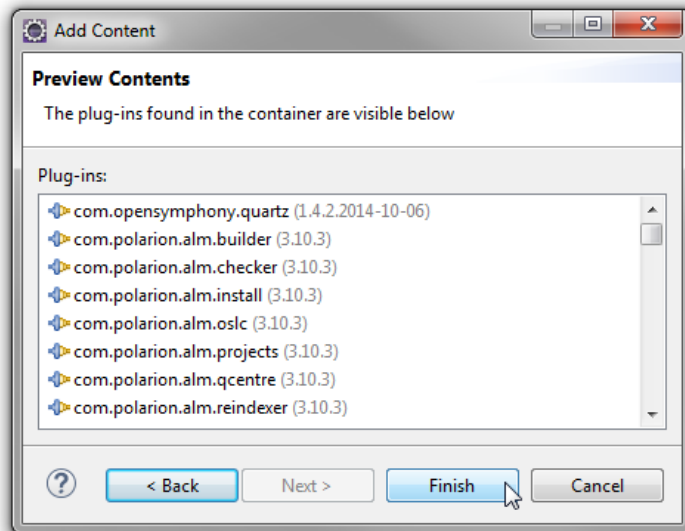


Figure WP-2: Currently Installed Polaron Plug-ins

8. A list of currently installed Polaron plug-ins appears. Click **Finish**.

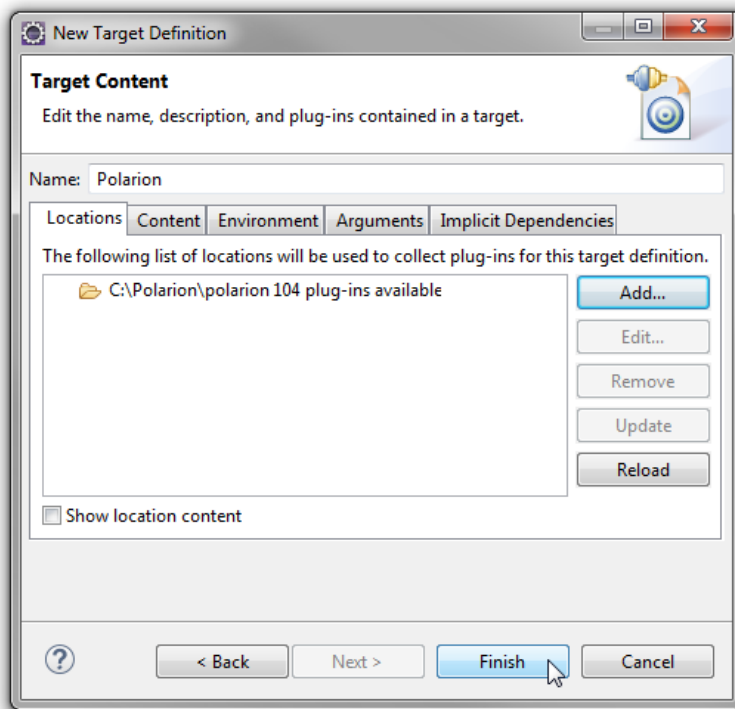


Figure WP-2: Confirm the Selected Path

9. The selected path and the number of discovered plug-ins available appear. Confirm that the path is correct and click **Finish**.

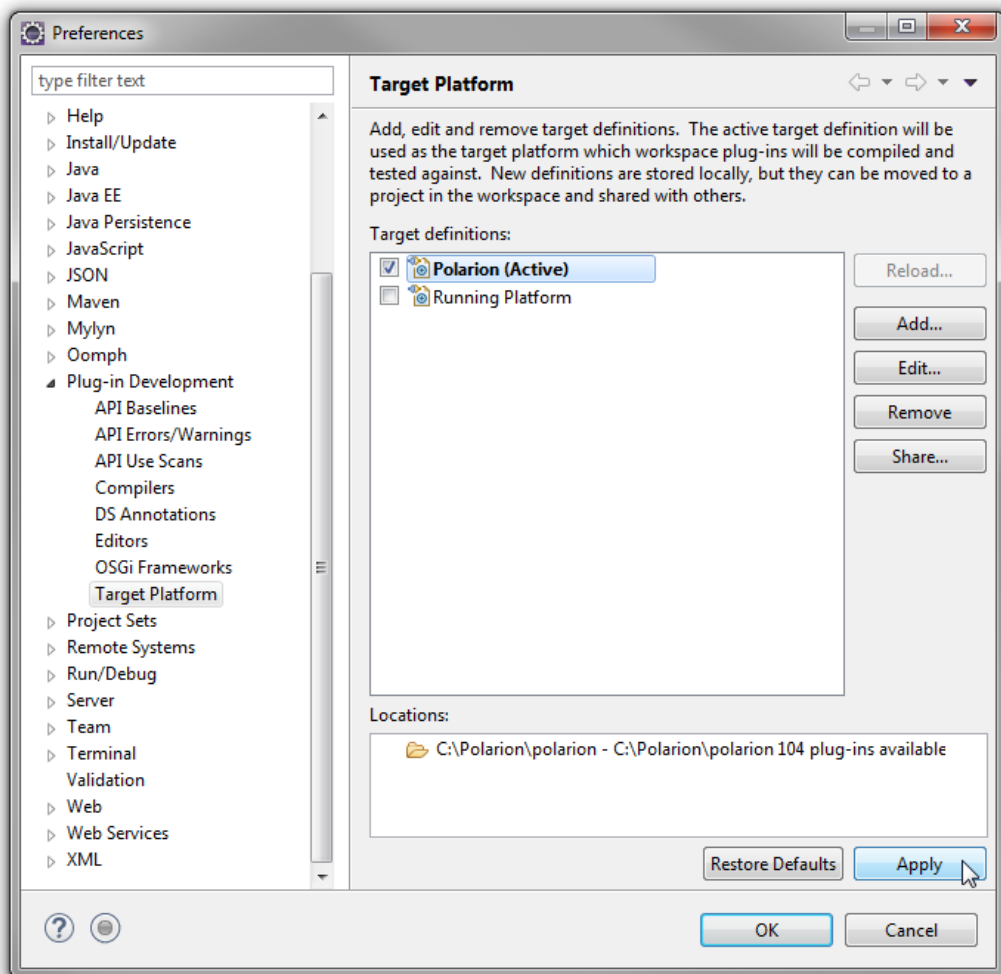


Figure WP-2: Select the Target Platform

10. Check the box beside the newly added path and click **Apply**.

Deployment to Installed Polarion

You can deploy a plugin to Polarion in two ways. First you can export a project as **Deployable Plugins and Fragments**. The second

way is described in the following section *Execution from Workspace*. To export the plug-in, perform these steps:

1. Select **File > Export...**
2. In the dialog that appears, select **Deployable Plugins and Fragments** in **Plug-in Development** section and click the **Next** button.
3. Mark your project (e.g. for **Servlet** example it will be `com.polarion.example.servlet`), and as the destination directory specify the `polarion` folder of your Polarion installation directory (usually in `C:\Polarion\polarion`)
4. At the **Options** card be sure, that **Package plug-ins as individual JAR archives** is unchecked. Click *Finish*.
5. Because this is a new polarion plug-in extension, you have to restart your Polarion server.

Execution from Workspace

The second way to deploy the plug-in to Polarion is to launch Polarion directly from your Eclipse workspace. This method has the added advantage of debugging the code directly in Eclipse.

1. Select **Run > Open Debug Configurations..**
2. Create a new Eclipse application (double click on *Eclipse Application*)
3. You should set:

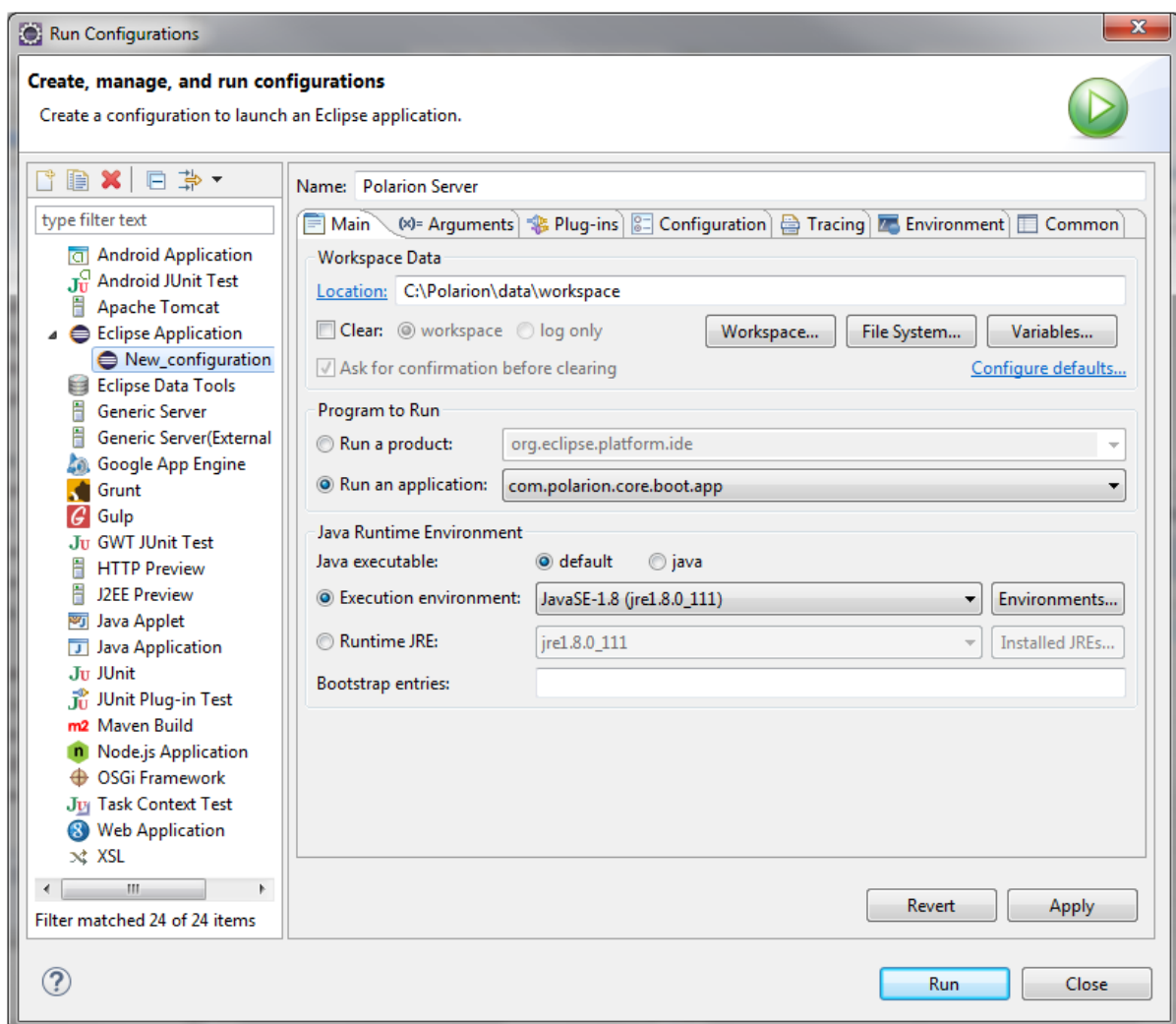


Figure WP-2: Debug - Main page

- **Name** to Polarion Server
 - **Workspace Data Location** to `C:\Polarion\data\workspace` (assuming that your Polarion is installed in `C:\Polarion\`)
 - **Run an application** to `com.polarion.core.boot.app` in the **Program to Run** section
 - Finally set your Runtime JRE
4. On the second, "**Program Arguments**" tab, set the following arguments:
 - to **Program Arguments** section:

```
-os win32 -ws win32 -arch x86 -appId polarion.server
```

1.

- to **VM Arguments** section:

```
-Xms256m -Xmx640m -XX:MaxPermSize=128m -XX:PermSize=128m  
-Xbootclasspath/a:C:\Polarion\bundled\java\lib\tools.jar  
-Dcom.polarion.home=C:\Polarion\polarion
```

Of course, you have to change parameters to Polarion server according your installation.

Note that, when using Java 8 the arguments `XX:MaxPermSize` and `XX:PermSize` are obsolete and must not be used anymore.

1.

- You can check the settings with the following screenshot:

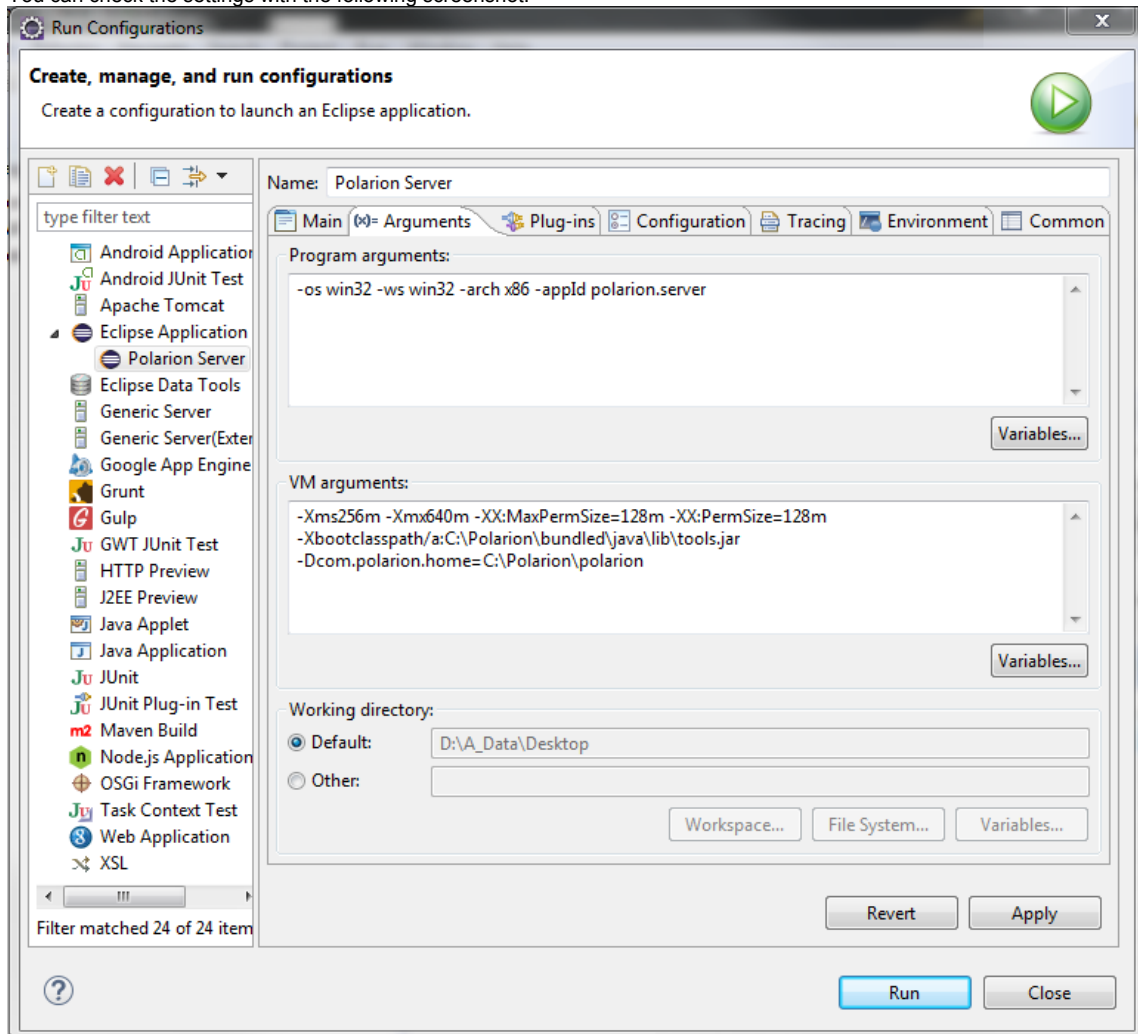


Figure WP-3: Debug - Arguments page

2. On the third "**Plugins**" tab, make sure, you have also selected "**Target Platform**" plugins.
3. Select all, and then click the **Validate Plug-ins** button. If there are some problems, uncheck the plugins which are in conflict.

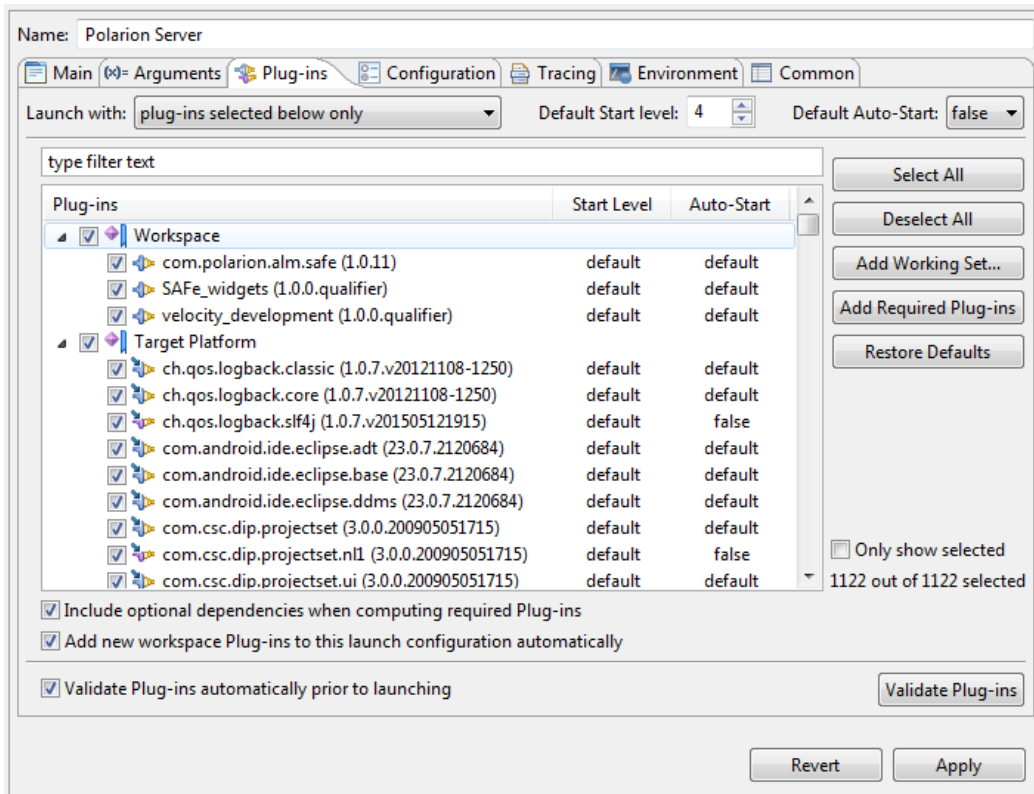


Figure WP-4: Debug - Plug-ins page

- Other pages shouldn't be changed. Just click the **Debug** button, and go on with your new Polarion Server application.