Custom Exporter Example

CEE - Introduction 1
CEE - Development Environment 1
CEE - Workspace Preparation 1
CEE - Execution 1
CEE - Hivemind Parameters 1
CEE - Custom Export Fields 1

See also the main SDK page.

CEE - Introduction

This exporter lets you test the export of Work Items into a JSON file.

CEE - Development Environment

See section *Requirements*.

CEE - Workspace Preparation

If you do not yet a have a workspace prepared for Polarion plugin development, please see sections *Workspace preparation* before proceeding with this chapter.

You must first import the **com.polarion.example.exporter** project from the `SDK\example` directory.
(`Polarion\polarion\SDK\examples\com.polarion.example.exporter` by default.)

Import the com.polarion.example.exporter project:

1. Start **Eclipse**, then select **File > Import...**
2. Select **Existing Projects into Workspace** in the **General** section of the dialog that appears.
3. Click **Next**.
4. Click **Browse** and select the **com.polarion.example.exporter** directory.
5. Click **OK**.
6. Select the **com.polarion.example.exporter** project from the **Projects** subwindow and click **Finish**.

CEE - Execution

To see how to run/debug Polarion with the example plugin see sections *Deployment to Installed Polarion* and *Execution from Workspace*.

Once Polarion is running, execute the following steps to see how the Custom Exporter Example works:

1. Select a **Work Item** in Polarion from the
   Tree or
   Table views.
2. Click on the
   icon in the toolbar.
3. Click
   Export.
4. Open the **Format** drop-down list.
5. If the plugin import is successful, you will see the option **json: Example Exporter** among the items of the list.
6. If the option fails to appear in the drop-down list, double-check if the `com.polarion.example.exporter` is among the selected plug-ins in the Run/Debug options of your IDE.
7. Select **json: Example Exporter**.
8. *(Optional)* Click **Show Advanced options** to view the exporter's custom parameters.
9. Click **OK**.

CEE - Hivemind Parameters

This sample code showcases how Hivemind parameters can be defined and used in the code. In this example, I have created 4 custom parameters: **max_workitems_for_export**, **write_result_in_german**, **english_result_message** and **german_result_message**, which are defined in the hivemodule.xml of the plugin and are of the type Integer, Boolean, String and String respectively. All of these types are supported and the usage of the parameters in the code can be seen in different parts of the `ExampleExporterCommand.java` class. The parameter **max_workitems_for_export** limits the number of Work Items exported. The effect of the **write_result_in_german** parameter can be seen in the text shown in the export dialog once the export has successfully ended. If the parameter is set to true, the final message shown there will be in German (as defined in the **german_result_message** parameter), otherwise it will be shown in English (as defined in the **english_result_message** parameter).

Info: The `contentType` in the hivemodule's `exporterDescriptor` section is the MIME type of the exported content.

CEE - Custom Export Fields

If you are not satisfied with the kind of methods and capabilities the default export fields in Polarion exporters have, you can create your own custom export field. An example of that can be seen in the examples `CustomExporterField.java` class. It implements the `IExportField` API interface and is given additional methods to set the `readOnly` and `columnWidth` fields. The former is then used in `ExampleExporterCommand.java` to apply the read-only status to fields based on own preferences and needs.

Requirements

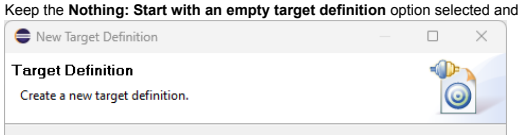
Development Environments

- Eclipse IDE for Enterprise Java 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)
- Eclipse Temurin™ 17 (LTS) by Adoptium for building and running your code.

Workspace Preparation

To start developing a Polarion 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**.



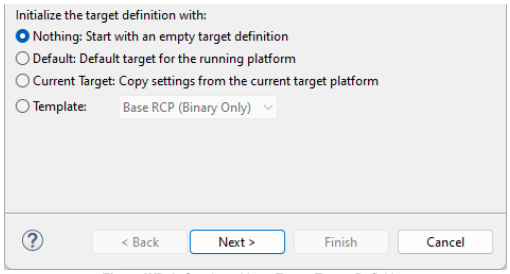


Figure WP-1: Starting with an Empty Target Definition

5. Enter a **Name** and click **Add**.
6. Select **Directory** and click **Next**.
7. Click **Browse** and select the `C:\Polarion\polarion` folder *(Windows)* or `/opt/polarion/polarion` *(Linux)*. (One level above the *plugins* folder.)
8. Click **Next**.

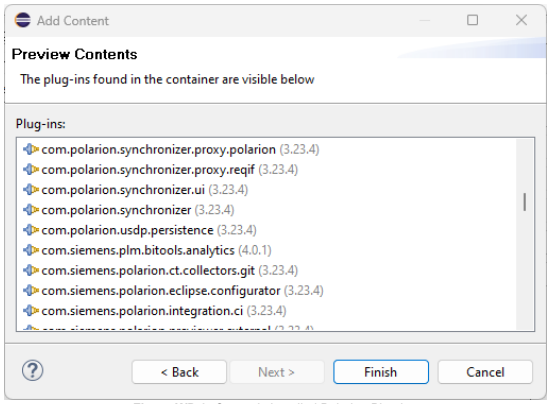


Figure WP-2: Currently Installed Polarion Plug-ins

9. A list of currently installed Polarion plug-ins appears. Click **Finish**.

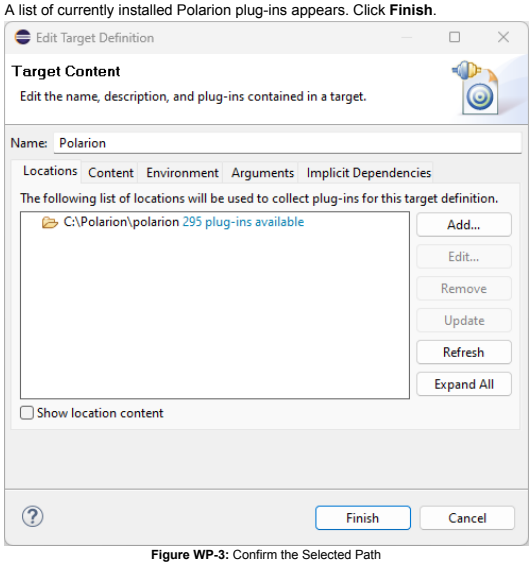


Figure WP-3: Confirm the Selected Path

10. The selected path and the number of discovered plug-ins available appear. Confirm that the path is correct and click **Finish**.{WP_4.png|description=**Figure WP-4:** Select the Target Platform}
11. 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.

NOTE: Servlets loaded by Polarion are cached in: `[Polarion_Home]\data\workspace\.config`. If this folder is not deleted before deploying a servlet extension (plugin) and restarting Polarion, then either the servlets will not be properly loaded, or the old ones will be loaded.

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:

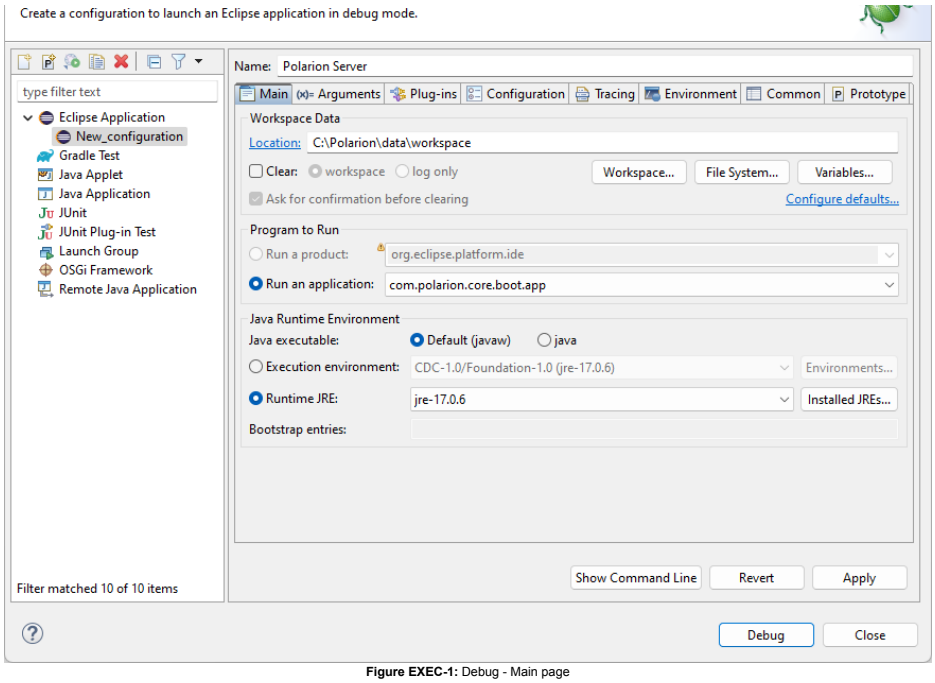


Figure EXEC-1: Debug - Main page

- **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.

4. Finally set your Runtime JRE. On the second, "**Arguments**" tab, set the following arguments:

In the **Program Arguments** section:

Windows:

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

Linux:

```
-os linux -ws gtk -arch x86_64 -appId polarion.server
```

In the **VM Arguments** section:

Windows:

```
-Xms1g -Xmx1g
-Dcom.polarion.home=C:\Polarion\polarion
-XX:+UseBiasedLocking -XX:BiasedLockingStartupDelay=0
--add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.annotation=ALL-UNNAMED
--add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.module=ALL-UNNAMED --add-opens=java.base/java.lang.ref=ALL-UNNAMED
--add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.math=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED
--add-opens=java.base/java.net.spi=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.nio.channels=ALL-UNNAMED
--add-opens=java.base/java.nio.channels.spi=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-UNNAMED --add-opens=java.base/java.nio.charset.spi=ALL-UNNAMED
--add-opens=java.base/java.nio.file=ALL-UNNAMED --add-opens=java.base/java.nio.file.attribute=ALL-UNNAMED --add-opens=java.base/java.nio.file.spi=ALL-UNNAMED
--add-opens=java.base/java.security=ALL-UNNAMED --add-opens=java.base/java.security.cert=ALL-UNNAMED --add-opens=java.base/java.security.interfaces=ALL-UNNAMED
--add-opens=java.base/java.security.spec=ALL-UNNAMED --add-opens=java.base/java.text=ALL-UNNAMED --add-opens=java.base/java.text.spi=ALL-UNNAMED
--add-opens=java.base/java.time=ALL-UNNAMED --add-opens=java.base/java.time.chrono=ALL-UNNAMED --add-opens=java.base/java.time.format=ALL-UNNAMED
--add-opens=java.base/java.time.temporal=ALL-UNNAMED --add-opens=java.base/java.time.zone=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED
--add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.base/java.util.concurrent.atomic=ALL-UNNAMED --add-opens=java.base/java.util.concurrent.locks=ALL-UNNAMED
--add-opens=java.base/java.util.function=ALL-UNNAMED --add-opens=java.base/java.util.jar=ALL-UNNAMED --add-opens=java.base/java.util.regex=ALL-UNNAMED
--add-opens=java.base/java.util.spi=ALL-UNNAMED --add-opens=java.base/java.util.stream=ALL-UNNAMED --add-opens=java.base/java.util.zip=ALL-UNNAMED
--add-opens=java.base/sun.nio.fs=ALL-UNNAMED --add-opens=java.base/sun.security.ssl=ALL-UNNAMED --add-opens=java.datatransfer/java.awt.datatransfer=ALL-UNNAMED
--add-opens=java.desktop/java.applet=ALL-UNNAMED --add-opens=java.desktop/java.awt=ALL-UNNAMED --add-opens=java.desktop/java.awt.color=ALL-UNNAMED
--add-opens=java.desktop/java.awt.desktop=ALL-UNNAMED --add-opens=java.desktop/java.awt.dnd=ALL-UNNAMED --add-opens=java.desktop/java.awt.dnd.peer=ALL-UNNAMED
--add-opens=java.desktop/java.awt.event=ALL-UNNAMED --add-opens=java.desktop/java.awt.font=ALL-UNNAMED --add-opens=java.desktop/java.awt.geom=ALL-UNNAMED
--add-opens=java.desktop/java.awt.im=ALL-UNNAMED --add-opens=java.desktop/java.awt.im.spi=ALL-UNNAMED --add-opens=java.desktop/java.awt.image=ALL-UNNAMED
--add-opens=java.desktop/java.awt.image.renderable=ALL-UNNAMED --add-opens=java.desktop/java.awt.peer=ALL-UNNAMED --add-opens=java.desktop/java.awt.print=ALL-UNNAMED
--add-opens=java.desktop/java.beans=ALL-UNNAMED --add-opens=java.desktop/java.beans.beancontext=ALL-UNNAMED --add-opens=java.desktop/javax.swing=ALL-UNNAMED
--add-opens=java.desktop/javax.swing.border=ALL-UNNAMED --add-opens=java.desktop/javax.swing.text=ALL-UNNAMED --add-opens=java.desktop/javax.swing.text.html=ALL-UNNAMED
--add-opens=java.desktop/sun.awt=ALL-UNNAMED --add-opens=java.desktop/sun.font=ALL-UNNAMED --add-opens=java.desktop/sun.java2d=ALL-UNNAMED
--add-opens=java.instrument/java.lang.instrument=ALL-UNNAMED --add-opens=java.logging/java.util.logging=ALL-UNNAMED --add-opens=java.management/java.lang.management=ALL-UNNAMED
--add-opens=java.prefs/java.util.prefs=ALL-UNNAMED --add-opens=java.rmi/java.rmi=ALL-UNNAMED --add-opens=java.rmi/java.rmi.dgc=ALL-UNNAMED
--add-opens=java.rmi/java.rmi.registry=ALL-UNNAMED --add-opens=java.rmi/java.rmi.server=ALL-UNNAMED --add-opens=java.sql/java.sql=ALL-UNNAMED
--add-opens=java.xml/com.sun.org.apache.xerces.internal.dom=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.jaxp=ALL-UNNAMED
--add-opens=java.xml/com.sun.org.apache.xerces.internal.parsers=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.util=ALL-UNNAMED
```

Linux

```
-Xms1g -Xmx1g
-Dcom.polarion.home=/opt/polarion/polarion -Dcom.polarion.propertyFile=/opt/polarion/etc/polarion.properties
-XX:+UseBiasedLocking -XX:BiasedLockingStartupDelay=0
--add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.annotation=ALL-UNNAMED
--add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.module=ALL-UNNAMED --add-opens=java.base/java.lang.ref=ALL-UNNAMED
--add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.math=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED
--add-opens=java.base/java.net.spi=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.nio.channels=ALL-UNNAMED
--add-opens=java.base/java.nio.channels.spi=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-UNNAMED --add-opens=java.base/java.nio.charset.spi=ALL-UNNAMED
--add-opens=java.base/java.nio.file=ALL-UNNAMED --add-opens=java.base/java.nio.file.attribute=ALL-UNNAMED --add-opens=java.base/java.nio.file.spi=ALL-UNNAMED
--add-opens=java.base/java.security=ALL-UNNAMED --add-opens=java.base/java.security.cert=ALL-UNNAMED --add-opens=java.base/java.security.interfaces=ALL-UNNAMED
--add-opens=java.base/java.security.spec=ALL-UNNAMED --add-opens=java.base/java.text=ALL-UNNAMED --add-opens=java.base/java.text.spi=ALL-UNNAMED
--add-opens=java.base/java.time=ALL-UNNAMED --add-opens=java.base/java.time.chrono=ALL-UNNAMED --add-opens=java.base/java.time.format=ALL-UNNAMED
--add-opens=java.base/java.time.temporal=ALL-UNNAMED --add-opens=java.base/java.time.zone=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED
--add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.base/java.util.concurrent.atomic=ALL-UNNAMED --add-opens=java.base/java.util.concurrent.locks=ALL-UNNAMED
--add-opens=java.base/java.util.function=ALL-UNNAMED --add-opens=java.base/java.util.jar=ALL-UNNAMED --add-opens=java.base/java.util.regex=ALL-UNNAMED
--add-opens=java.base/java.util.spi=ALL-UNNAMED --add-opens=java.base/java.util.stream=ALL-UNNAMED --add-opens=java.base/java.util.zip=ALL-UNNAMED
--add-opens=java.base/sun.nio.fs=ALL-UNNAMED --add-opens=java.base/sun.security.ssl=ALL-UNNAMED --add-opens=java.datatransfer/java.awt.datatransfer=ALL-UNNAMED
--add-opens=java.desktop/java.applet=ALL-UNNAMED --add-opens=java.desktop/java.awt=ALL-UNNAMED --add-opens=java.desktop/java.awt.color=ALL-UNNAMED
--add-opens=java.desktop/java.awt.desktop=ALL-UNNAMED --add-opens=java.desktop/java.awt.dnd=ALL-UNNAMED --add-opens=java.desktop/java.awt.dnd.peer=ALL-UNNAMED
--add-opens=java.desktop/java.awt.event=ALL-UNNAMED --add-opens=java.desktop/java.awt.font=ALL-UNNAMED --add-opens=java.desktop/java.awt.geom=ALL-UNNAMED
--add-opens=java.desktop/java.awt.im=ALL-UNNAMED --add-opens=java.desktop/java.awt.im.spi=ALL-UNNAMED --add-opens=java.desktop/java.awt.image=ALL-UNNAMED
--add-opens=java.desktop/java.awt.image.renderable=ALL-UNNAMED --add-opens=java.desktop/java.awt.peer=ALL-UNNAMED --add-opens=java.desktop/java.awt.print=ALL-UNNAMED
```

```
--add-opens=java.desktop/java.beans=ALL-UNNAMED --add-opens=java.desktop/java.beans.beancontext=ALL-UNNAMED --add-opens=java.desktop/javax.swing=ALL-UNNAMED
--add-opens=java.desktop/javax.swing.border=ALL-UNNAMED --add-opens=java.desktop/javax.swing.text=ALL-UNNAMED --add-opens=java.desktop/javax.swing.text.html=ALL-UNNAMED
--add-opens=java.desktop/sun.awt=ALL-UNNAMED --add-opens=java.desktop/sun.font=ALL-UNNAMED --add-opens=java.desktop/sun.java2d=ALL-UNNAMED
--add-opens=java.instrument/java.lang.instrument=ALL-UNNAMED --add-opens=java.logging/java.util.logging=ALL-UNNAMED --add-opens=java.management/java.lang.management=ALL-UNNAMED
--add-opens=java.prefs/java.util.prefs=ALL-UNNAMED --add-opens=java.rmi/java.rmi=ALL-UNNAMED --add-opens=java.rmi/java.rmi.dgc=ALL-UNNAMED
--add-opens=java.rmi/java.rmi.registry=ALL-UNNAMED --add-opens=java.rmi/java.rmi.server=ALL-UNNAMED --add-opens=java.sql/java.sql=ALL-UNNAMED
--add-opens=java.xml/com.sun.org.apache.xerces.internal.dom=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.jaxp=ALL-UNNAMED
--add-opens=java.xml/com.sun.org.apache.xerces.internal.parsers=ALL-UNNAMED --add-opens=java.xml/com.sun.org.apache.xerces.internal.util=ALL-UNNAMED
```

5. You must now change the parameters to the Polarion server based on your installation. You can check the settings with the following screenshot:

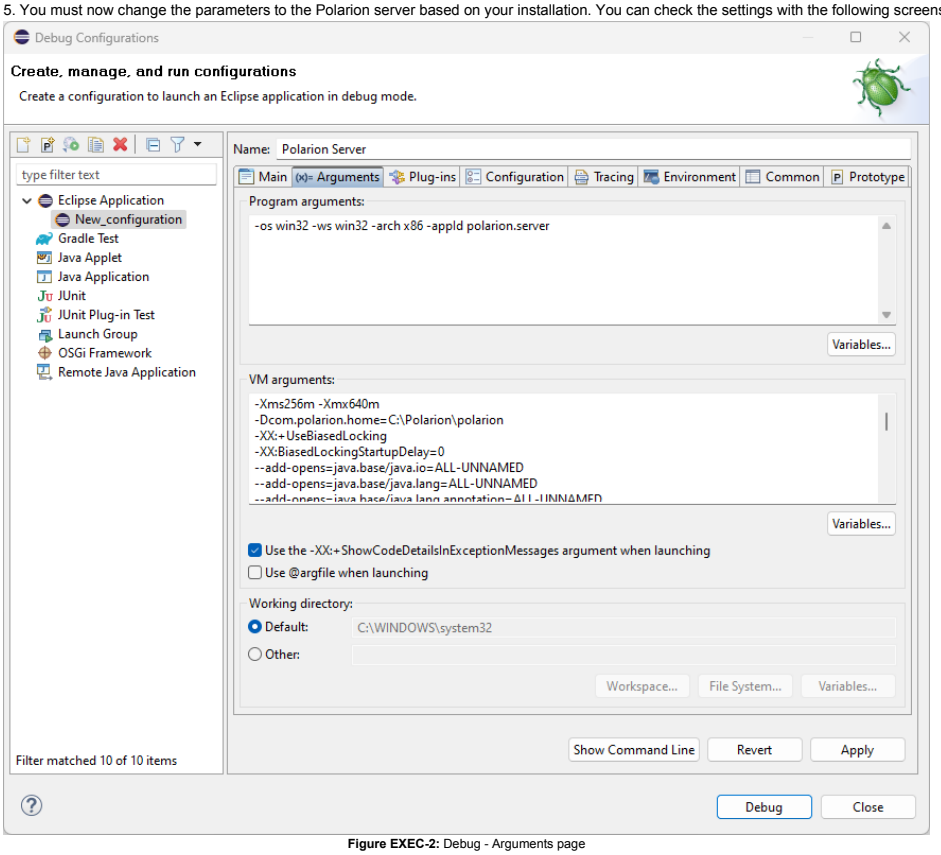


Figure EXEC-2: Debug - Arguments page

6. On the third "**Plug-ins**" tab, make sure, you have also selected "**Target Platform**" plugins.

7. Select all, and then click the **Validate Plug-ins** button. If there are some problems, uncheck the plugins which are in conflict.

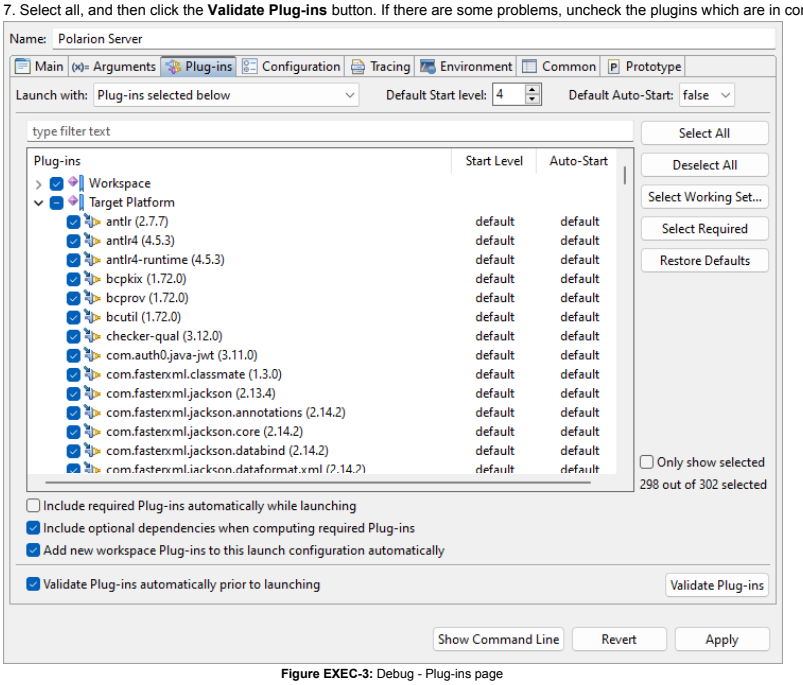


Figure EXEC-3: Debug - Plug-ins page

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