

Jaspersoft® Studio Source Build Guide

Software Release 8.2.0



Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, Jaspersoft, JasperReports, and Visualize.js are registered trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of Cloud Software Group, Inc. may be covered by registered patents. Please refer to Cloud Software Group's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2005-2023. Cloud Software Group, Inc. All Rights Reserved.

TABLE OF CONTENTS

Chapter 1 Jaspersoft Studio Source Build	5
1.1 Zip Package Contents	5
1.2 Setting Up Eclipse and Your Workspace	5
1.3 Setting Up Run/Debug Configuration of Jaspersoft Studio	10
1.4 Creating Products from Jaspersoft Studio Sources	12

CHAPTER 1 JASPERSOFT STUDIO SOURCE BUILD

This document describes how to compile and run Jaspersoft Studio Professional edition from the source files. Version 8.2.0 is used as example, but the details are also valid for newer versions.

1.1 Zip Package Contents

The zip package named `TIB_js-jss_X.Y.Z_sources.zip` contains the following items:

- `jrePackages` folder: Contains zip files of the JRE version shipped with Jaspersoft® Studio (needed during build time).
- `sources` folder: Contains the source code of the Jaspersoft Studio Professional version.
- `targetRepoE426` folder: Contains the target repository that can be used for both compiling in the Eclipse and building via Maven Tycho.
- `TIB_js-jss_8.2.0_license.txt` folder: The current license file.

1.2 Setting Up Eclipse and Your Workspace

To set up Eclipse:

1. Download the latest version of Eclipse from the website: <http://www.eclipse.org/>.
Normally, the Jaspersoft® developers use the **Eclipse IDE for RCP and RAP developers** version of Eclipse. As of the time this document was created, the latest version is Eclipse 2023-06(4.28).
2. Choose a blank workspace when starting Eclipse instance.
3. Configure the proper Java Virtual Machine for compiling Jaspersoft Studio source code:
 - a. Select **Eclipse > Preferences > Java > Installed JREs**. Make sure that Java 11 and 8 is in the list of items. This is because the Jaspersoft Studio source code is still compatible with version 8, but the final product runs on 11. Therefore, it is better to have all of them set up properly (<https://adoptium.net/download/>).

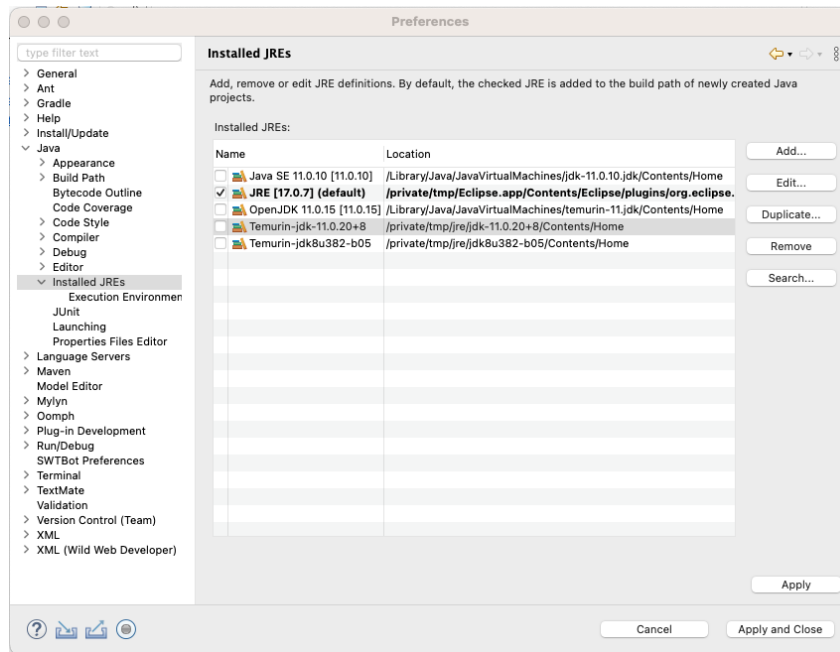


Figure 1-1 Installed JREs configuration

- b. Select **Eclipse > Preferences > Java > Installed JREs > Execution Environment**. Remember to select the best compatible JVM for each execution environment(Java 8 and 11).

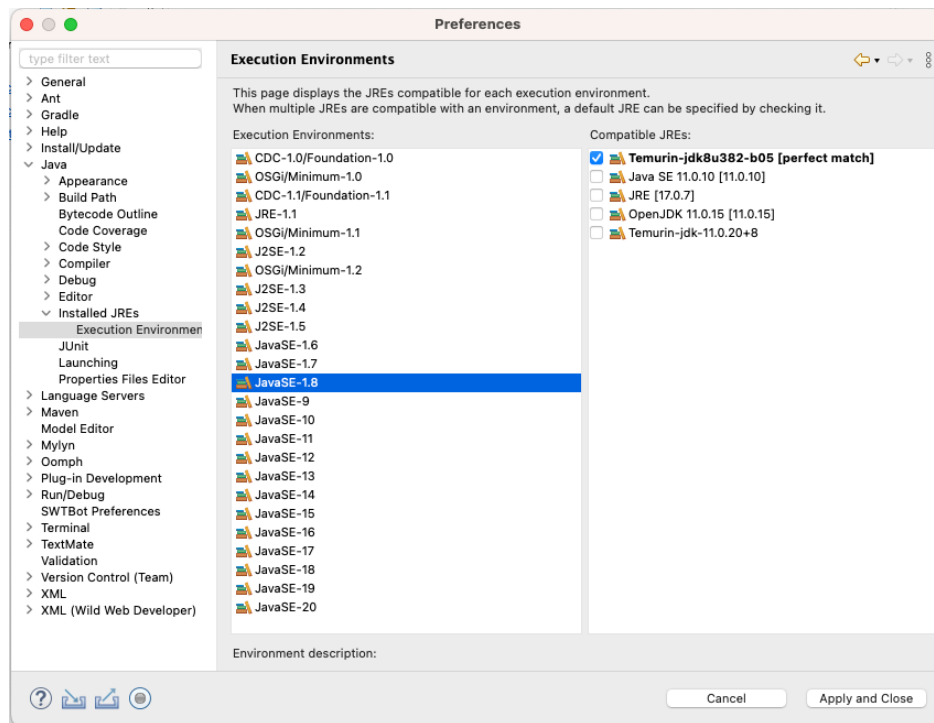


Figure 1-2 Execution Environments Configuration

- c. Select **Eclipse > Preferences > Java > Compiler**. Remember to change the Compiler compliance level to 1.8.

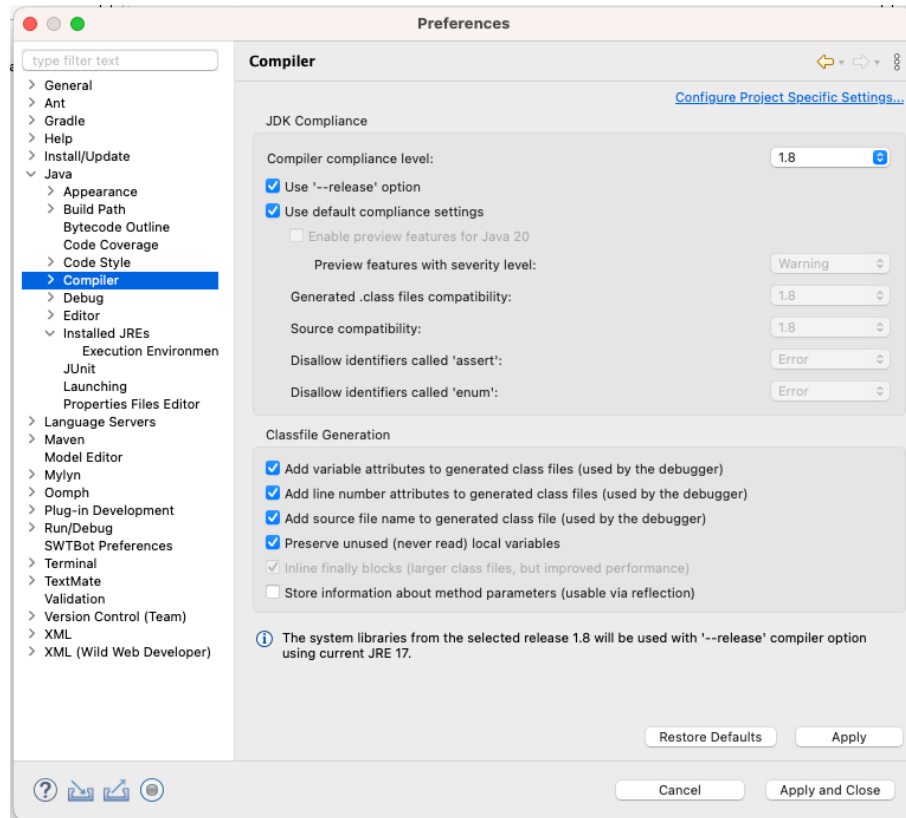


Figure 1-3 Compiler Configuration

4. Configure the proper target platform for compiling Jaspersoft Studio source code:
 - a. Select **Eclipse > Preferences > Plug-in Development > Target Platform**.
 - b. Add new platform starting from the **Nothing** option, then add the targetRepoE426-8.2.0 directory containing the plug-ins and features that are the foundation for the Jaspersoft Studio platform.

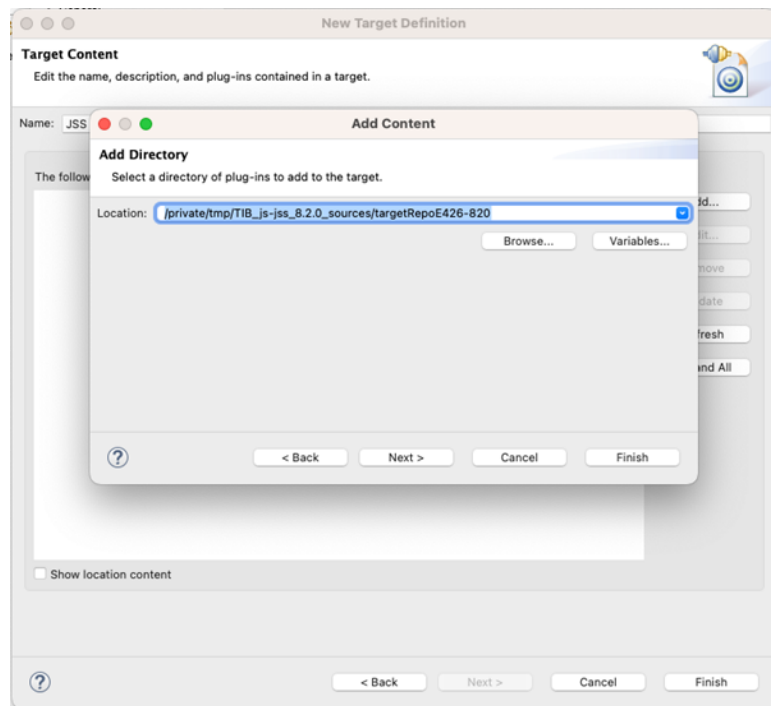


Figure 1-4 Creating a target platform

- c. Select the newly created platform to compile the sources.

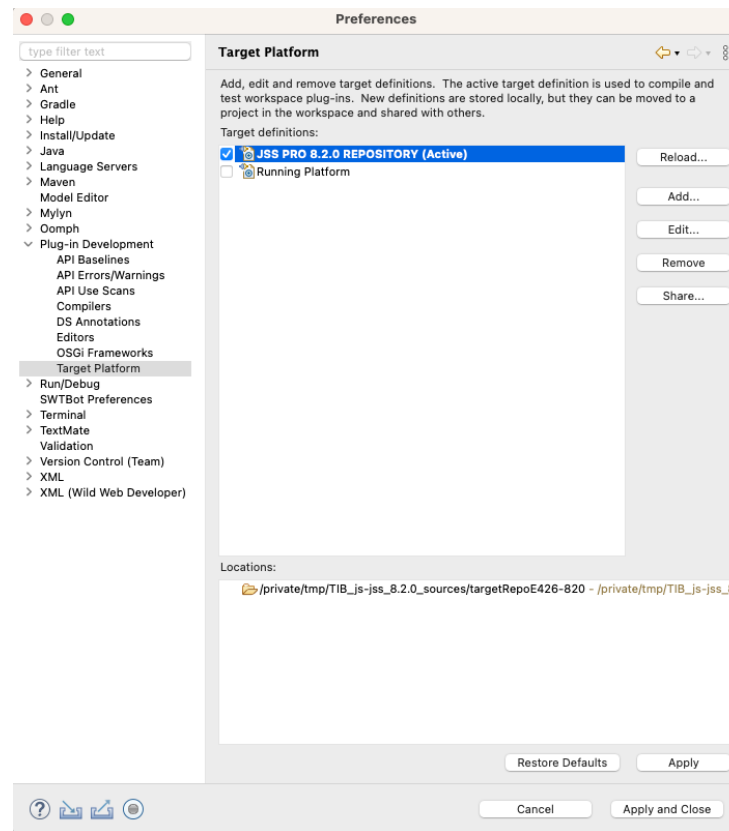


Figure 1-5 Selecting the target platform

5. Import the existing projects from the sources folder:
 - a. Select **File > Import**.
 - b. Select **Import (wizard) > General > Existing Projects into Workspace**.

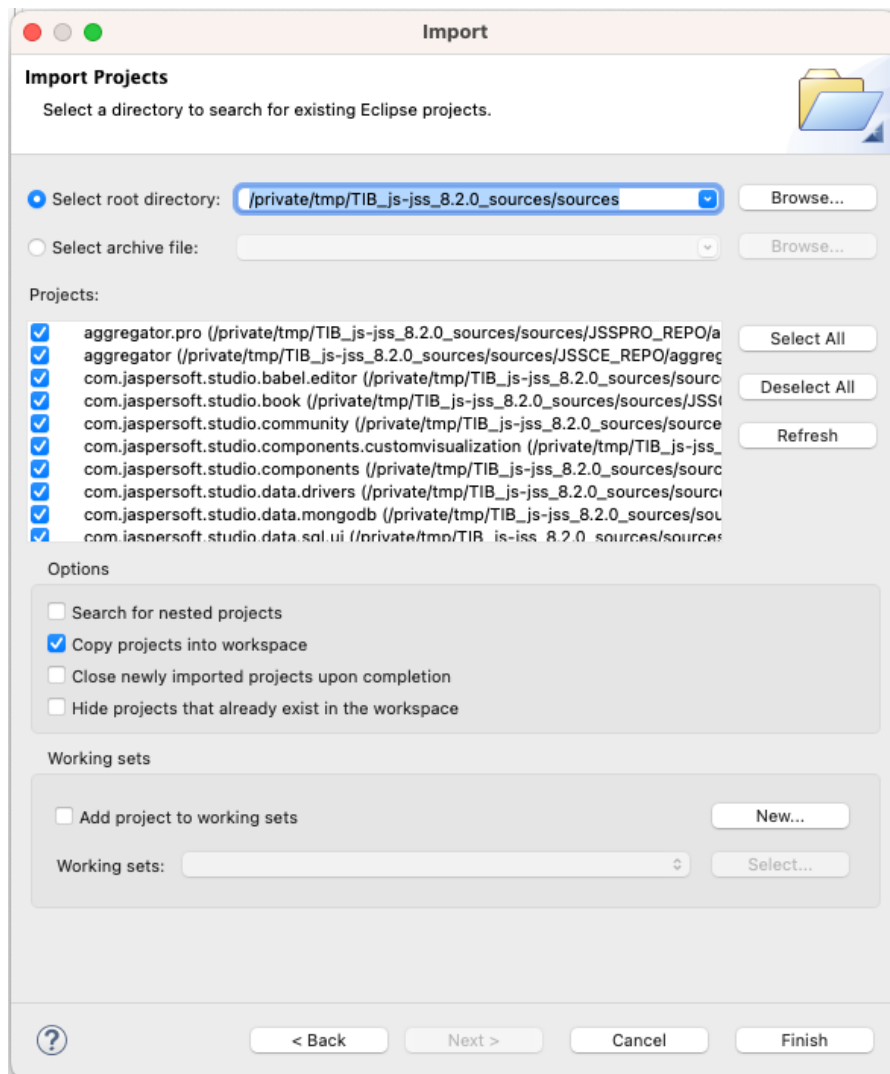


Figure 1-6 Importing projects

- c. Configure the wizard to point to the location of the downloaded sources directory, then click **Finish**. It is recommended to copy the projects directly into the dedicated workspace to avoid modifying the original ones.



Once the import operation ends, you see the automatic build triggered. The projects `org.osgi.framework.system.packages.extra` and `com.jaspersoft.studio.rcp.macosx` (if not on Mac) could appear as broken. You can safely close them.

1.3 Setting Up Run/Debug Configuration of Jaspersoft Studio

To run/debug a version of the Jaspersoft Studio runtime from the development environment, you can create a run/debug configuration:

1. Select **Run > Debug Configurations > Eclipse Application > New** from the menu.
2. Enter the name and select `com.jaspersoft.studio.pro.rcp.product` for **Run a product** option.
3. Set the **Execution environment** to the Java 11 version.

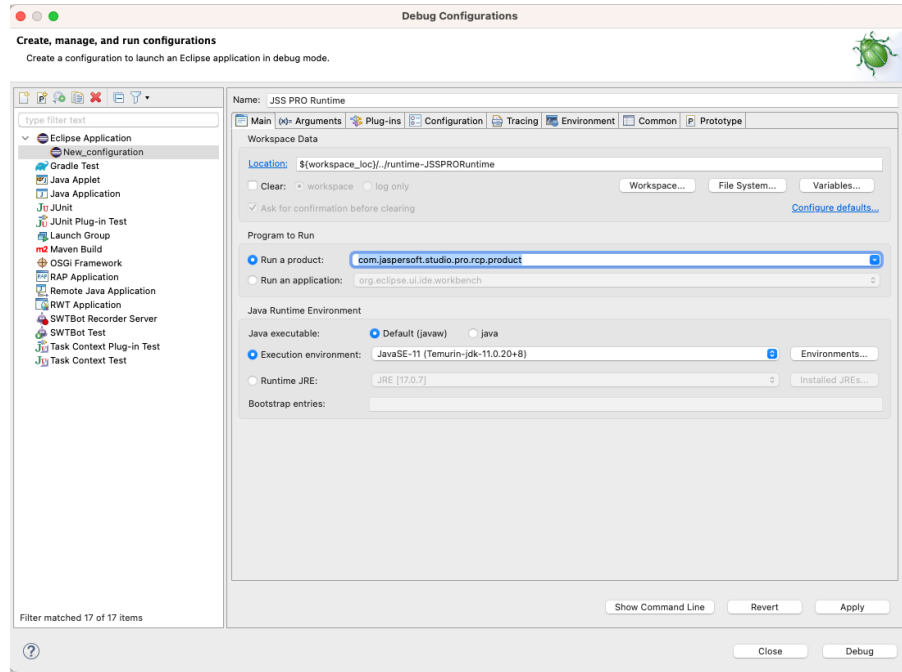


Figure 1-7 Creating a debug configuration

4. Enter the required details for the product configuration in the Main tab.
You can add the additional arguments and configuration parameters in the Arguments tab. You can also replicate the actual final product configuration using the information in the `jaspersoftstudiopro.product` file. The following example is based on the 8.2.0 configuration:

```
-Xms128m
-Xmx2048m
-XX:+CMSClassUnloadingEnabled
-XX:+UseConcMarkSweepGC
-Dfile.encoding=UTF-8
-Djava.net.preferIPv4Stack=true
-Ddeployment.security.TLSv1.1=true
-Ddeployment.security.TLSv1.2=true
-Djavax.xml.soap.MessageFactory=com.sun.xml.messaging.saaj.soap.ver1_1.SOAPMessageFactory1_1Impl
-Djavax.xml.soap.SOAPFactory=com.sun.xml.messaging.saaj.soap.ver1_1.SOAPFactory1_1Impl
-Djavax.xml.soap.SAAJMetaFactory=com.sun.xml.messaging.saaj.soap.SAAJMetaFactoryImpl
```

```
-Djavax.xml.soap.SOAPConnectionFactory=com.sun.xml.messaging.saaj.client.p2p.  
HttpSOAPConnectionFactory  
  
-XstartOnFirstThread  
  
-Dorg.eclipse.swt.graphics.Resource.reportNonDisposed=true
```

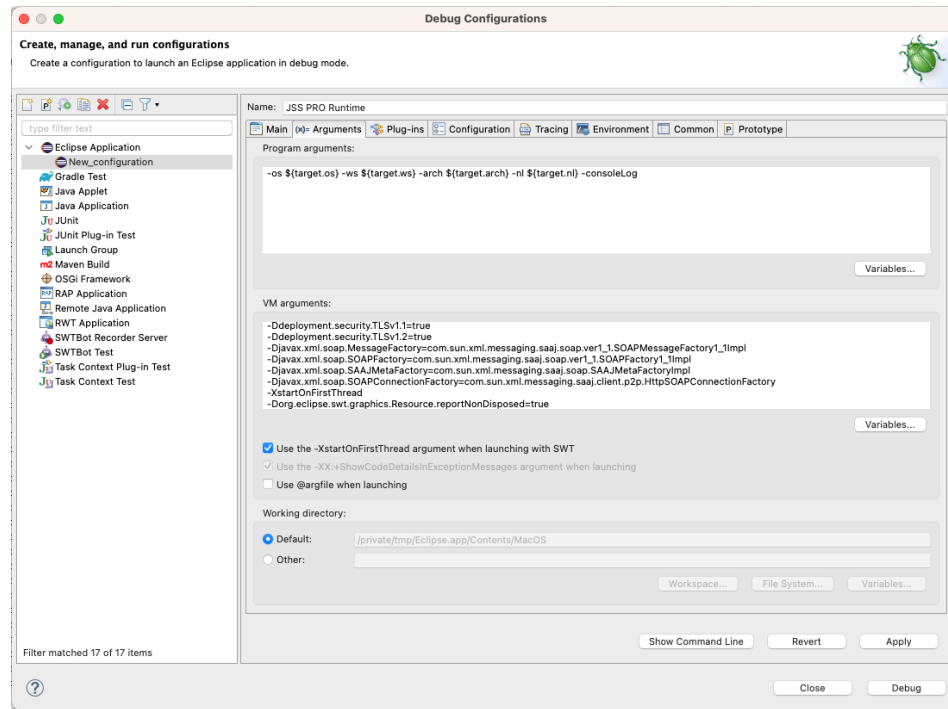


Figure 1-8 Setting arguments

5. (Optional) On the Plug-ins tab, disable the option **Validate Plug-ins automatically prior to launching** to avoid the warning message related to the plug-ins/fragments/features related to multiple operating systems.
6. Click the Debug/Run button to launch the Jaspersoft Studio runtime.

1.4 Creating Products from Jaspersoft Studio Sources

To test a final product as it looks after export, you can use Maven Tycho to build Jaspersoft Studio sources.



All the source projects must be in one single folder: for example, it could be the Eclipse workspace or copy the projects into a new dedicated folder.

- To build the Jaspersoft Studio Professional project, you need to use the `aggregator.pro` project as a main location.
 - The JasperReports® IO package for Jaspersoft Studio Sources is needed to perform a correct successful build. For example, in case of Jaspersoft Studio Professional 8.2.0, the correct version is 3.2.0: artifact `jrrio-standalone-3.2.0-jss.zip`.
1. Make sure you have maven correctly installed.

2. Set up Maven properly using any of these methods.

Method 1

- a. Configure your Maven `settings.xml` file. You can start from the following sample `settings.xml` file:

```
<settings>
  <profiles>
    <profile>
      <id>JSSProfile</id>
      <repositories>
        <repository>
          <id>central</id>
          <name>jaspersoft-repo</name>
          <url>https://jaspersoft.jfrog.io/jaspersoft/jaspersoft-repo</url>
        </repository>
      </repositories>
      <properties>
        <!-- Local Repository with the target platform for building -->
        <targetplatform.repo>/tmp/TIB_js-jss_8.2.0_sources/targetRepoE426-8.2.0
        </targetplatform.repo>
        <!-- JRE files location -->
        <jre.packages.location>/tmp/TIB_js-jss_8.2.0_sources/jrePackages
        </jre.packages.location>
        <!-- JRIO package location -->
        <jrio.package.location>/tmp/jrio</jrio.package.location>
        <!-- JRIO version -->
        <jrio.package.version>3.2.0</jrio.package.version>
      </properties>
    </profile>
  </profiles>
  <activeProfiles>
    <activeProfile>JSSProfile</activeProfile>
  </activeProfiles>
</settings>
```

- b. Inside the `aggregator.pro` project folder, enter the following command.

```
$ mvn clean package
```

OR Method 2: Provide all the details with one single command, once in the `aggregator.pro` folder:

```
mvn clean package \
-Dmaven.repo.remote=https://jaspersoft.jfrog.io/jaspersoft/jaspersoft-repo \
-Dtargetplatform.repo=/tmp/TIB_js-jss_8.2.0_sources/targetRepoE426-8.2.0 \
-Djre.packages.location=/tmp/TIB_js-jss_8.2.0_sources/jrePackages \
-Djrio.package.location=/tmp/jrio \
-Djrio.package.version=3.2.0
```

3. When the build is complete, you can retrieve the product artifacts from the following location:

```
<FULLWORKSPACE_PATH>/com.jaspersoft.studio.pro.rcp.product/target/products/
com.jaspersoft.studio.pro.rcp.product/
```

`FULLWORKSPACE_PATH` is the location where all the sources are located including an Eclipse workspace folder.

4. To properly package the previously created raw artifacts, perform the following tasks.

- a. Select the subfolder `ant-scripts` from `aggregator.pro` project as a current location.

- b. Run the following command using ANT:

```
ant -buildfile packageRawDistributions.xml -Djre.packages.location=/tmp/TIB_js-  
jss_8.2.0_sources/jrePackages
```
- c. In the project subfolder `product/dist`, you find the zipped artifacts ready to use.