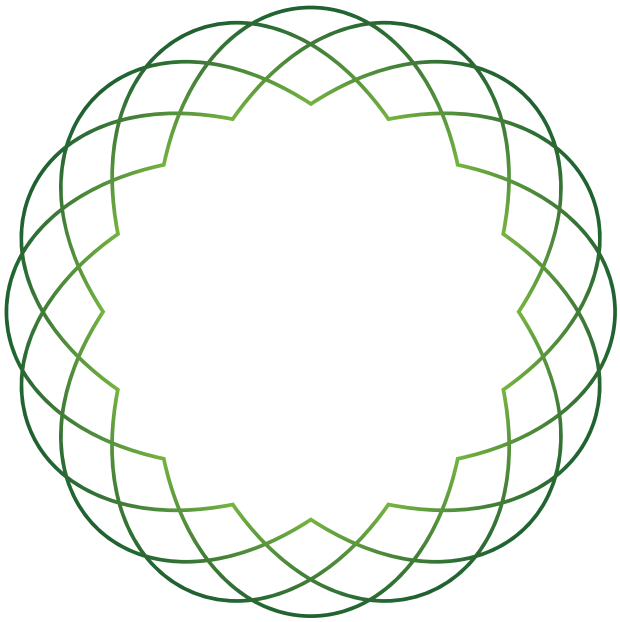


Predix Machine

Software Development Kit Installation Guide

Version 16.2

May 2016

Predix

Predix Machine

Contents

About this Guide

About This Guide

This guide is for developers who want to install the Predix Software Development Kit (SDK) to generate their own Predix Machine OSGi container.

For continuously updated installation and Predix Machine content, go to *https://predix.io/docs*.

Predix Machine SDK Overview

The Predix Machine Software Development Kit (SDK) is for developers who want to develop applications using Predix Machine. The Predix Machine SDK allows you to generate your own Predix Machine OSGi container by selecting certain feature groups that include all of the necessary bundles for that feature. You can also generate the container by selecting individual bundles. The grouping of the features is based on certain dependencies, as well as related features.

The following capabilities are included in all features, regardless of which of the following features you pick from the following list:

- OSGi Container
- Felix Dependency Manager
- Declarative Services Support
- Felix HTTP Bundle
- Jersey (jaxrs) Bundle
- MetaType-Configuration Management
- Logging
- Security Admin Service: SSL Certificate Management
- User Management Service (Account management)

The following table shows each of the Predix Machine bundles that are included in each of the Predix Machine feature groups.

Feature	Description	Dependency
Predix Application Services	• Git Repository Management Service	• Predix Web Tools
Predix Cloud Gateway	• Predix Cloud Identity Management	• Predix Web Tools
Predix Command Framework	• Sends commands to applications that are running on a device.	
Predix Device Management (Proximetry)	Allows the device to receive commands and updates from the cloud.	• Predix Web Tools • Predix Cloud Gateway • Command Framework • Predix Machine Provisioning Support (Proximetry and Enrollment)
Predix HTTP River	• Transfers data from Predix Machine-enabled edge devices to the Predix Cloud using HTTPS.	• Predix Web Tools • Predix Cloud Gateway

Feature	Description	Dependency
Predix HTTP Tunnel	• Facilitates communication of different network protocols through HTTP/HTTPS	• Predix Web Tools • Predix Cloud Gateway
Predix Machine Gateway	• OPC-UA Adapter • Modbus Adapter • Healthmonitor Adapter • Hoover	• Predix Web Tools • Predix Cloud Gateway • Store and Forward • HTTP River

Predix MQTT Support	• Publishes messages to a broker or subscribes to a topic to receive messages.
	  • MQTT River
	  • MQTT Client
		• Predix Web Tools
		• Predix Cloud Gateway

Predix OPC-UA Server	• Allows Predix Machine-enabled applications to expose data through the OPC-UA protocol, a common machine-to-machine protocol.	• Predix Web Tools
Predix Provisioning Support (Proximetry and Enrollment)	• Remote device management • Provisioning support when generating the container using scripts (PROV) • Bundles to handle ZIP support  **Note:** See *EdgeManager* for more information.	• Predix Web Tools • Predix Cloud Gateway • Command Framework
Predix Store and Forward	• Forwards data to the cloud and continuously stores data to prevent data loss.	
Predix Technician Console	• Predix Technician Console	• Predix Web Tools • Command Framework • Predix Cloud Gateway • Provision Support
Predix Web Console (Should only be used for debugging)	• Predix Machine Web Console and Bundle Updates	• Predix Web Tools

Feature	Description	Dependency
	Note: Should only be used for debugging	
Predix Web Tools	• JSON support with JAX-RS • HTTP services for REST support	
Predix WebSocket River	• WebSocket River	• Predix Web Tools • Predix Cloud Gateway • Predix WebSockets
Predix WebSockets	• WebSocket Server • WebSocket Client Service	
Predix Additional Services **Note:** Only available when using the Predix Machine SDK in Eclipse to generate the container.	• TCP Socket Server	

Downloading the Predix Machine SDK

You must have a Cloud Foundry account to access the download site.

To begin using the SDK, you must first download the SDK package.

1. Access the Predix Machine SDK download at *https://artifactory.predix.io/artifactory/PREDIX-EXT/predix-machine-package/predixmachinesdk/16.2.0/predixmachinesdk-16.2.0.zip*.
   You receive a limited product license key when you download the Predix Machine SDK. To obtain an unlimited license key, install Predix Machine on your device and complete the device-enrollment process in EdgeManager.
2. Download the `PredixMachineSDK-16.2.0.zip` file.
3. Unzip and extract all of the files in the ZIP file.

Predix Machine SDK Directory Structure

When extracted, the downloaded SDK file creates the following directory structure:

Directory	Description
`docs`	Contains the SDK documentation and the `apidocs.zip` file with the Javadoc APIs.
`eclipse-plugins`	Indicates the location that you will point the Eclipse installation to. Includes the following folders: • `features` • `plugins`
`license`	Contains the license files.
`samples`	Contains `sample-apps.zip` and `sample-cloud-apps.zip` files for sample applications.
`utilities`	Contains scripts used for generating containers.
`InstallationGuide.pdf`	The *Predix Machine Software Development Kit Installation Guide*.

Verifying Requirements

Requirements

Environment Variables

The following environment variable must be set for use by Predix Machine scripts:

`JAVA_HOME`: Java Virtual Machine location used with Predix Machine.

Note: Do not use a trailing backslash or add quotes to the `JAVA_HOME` environment variable.

Software Packages

The following software packages are required to use the Predix Machine SDK.

- Eclipse (Kepler, Luna, or Mars 64-bit versions) with the Plug-in Development Environment (PDE), for example Eclipse IDE for Java EE Developers.

  Note: Download Eclipse *Kepler*, *Luna*, or *Mars*.

  If you are using a different version for your development environment, you can install and use another Eclipse version to use for the Predix Machine SDK.

- Predix Machine SDK. Download the Predix Machine SDK .

  Note: See *Downloading the Predix Machine SDK* on page 9.

- Maven. Ensure that you have Maven installed. (On a command line interface, type `mvn -version`. Your version should be 3.1 or above.)

  Note: You can download Maven *here*.

Required Linux Utilities

Because certain devices may run minimal or limited versions of various operating systems, some utility executables that are used in various scripts to start and run the Predix Machine container may not be installed. The following utility executables are required for running Predix Machine in a Linux environment.

Utility	Description	Utility	Description
awk	Executes pattern-matching operations on data	**kill**	Sends a signal to a process
cat	Catenates	**nohup**	Prevents commands from aborting if you exit
chmod	Changes permissions of files or directories	**printf**	Prints a formatted string

Utility	Description	Utility	Description
command	Verifies if a command exists	**ps**	Displays process status
date	Prints or changes time and date	**readlink**	Prints value of symbolic link or file name
dirname	Strips last part of a filename	**sed**	Filters and transforms text
echo	Repeats typed text sent to a peripheral or	**sleep**	Delays or pauses
find	Searches text in a file	**sudo**	Allows you to execute commands as another user
getopts	Parses command line arguments	**systemctl**	systemd utility that controlls the systemd system and service manager.
grep	Processes text line by line and prints lines that match specified patterns	**tr**	Stops processor to wait for further instruction
head	Outputs first part of files	**trap**	Translates sets of characters
jar	Manipulates Java Archive (JAR) files	**uname**	Prints system information
java	Starts a Java application	**unzip**	Used to extract compressed files
keytool	Creates private keys		

Memory Requirements

Predix Machine does not provide memory management-related directives to the Java Virtual Machine (JVM), allowing the *Ergonomics* feature in the JVM to make intelligent choices that it can tune dynamically. The JVM makes these choices based on the class of the server Machine is installed on, which in turn is determined by the total available memory, the number of CPUs, and platform architecture (32-bit or 64-bit).

In the absence of explicit command line parameters specifying memory allocation, the JVM determines the minimum and maximum heap sizes at start-up, and ensures that the usage stays between these limits, growing and shrinking the committed heap allocation as necessary. For example, Java 7 and 8 set the minimum heap size to 1/64 of available physical memory, and the maximum heap size to 1/4 of available physical memory up to 1 GB, for a 32-bit system with two or more CPUs and 2 or more GB of RAM. The default maximum heap size can be up to 32 GB on a 64-bit system with 128 GB RAM or more.

Therefore, on a 64-bit Linux server with 64 GB RAM and 16 CPUs, Predix Machine (or any Java process that does not explicitly specify heap parameters) will be given a minimum heap size of 1GB and maximum heap size of 16 GB. On the other hand, Predix Machine running on a smaller device such as a Raspberry PI with 434 MB RAM will be given a minimum heap size of about 7 MB and maximum heap size of about 110 MB. Predix Machine has been found to operate well using the defaults on a variety of systems, including Raspberry PI.

Unless the situation demands otherwise, it is best practice to leave the heap configuration and tuning to the JVM. However, it is possible that the operating characteristics of application bundles running under Predix Machine may require more heap space than what is allocated by default. Heap space also depends on the features you selected in the Predix Machine container. Additionally, the heap usage may need to be reduced due to other applications running on the system.

Memory Footprint for Predix Machine Features

The following table shows the memory footprint of each of the Predix Machine Features.

Feature	Memory usage while container is running (MB)
Predix Additional Services	49
Predix Application Services	72.8
Predix Cloud Gateway	52.4
Predix Command Framework	17.2
Predix Device Management (Proximetry)	61.8
Predix HTTP River	54
Predix HTTP Tunnel	59
Predix Machine Gateway	36.5
Predix MQTT Support	34.5
Predix OPC-UA Server	188
Predix Provisioning (Proximetry and Enrollment)	70.3
Predix Store and Forward	106.2
Predix Technician Console	73.5
Predix Web Console	72.5
Predix Web Tools	58.1
Predix WebSocket River	63.4
Predix WebSockets	59.2

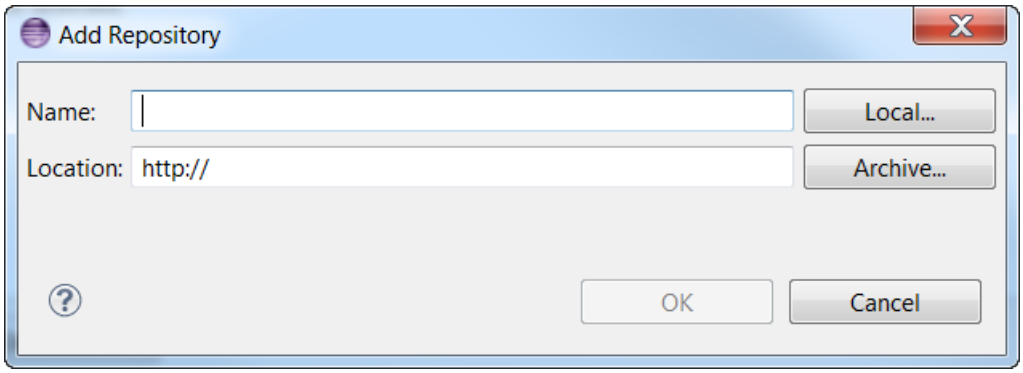
Installing the Predix Machine SDK

To install the Predix Machine SDK, follow these steps.

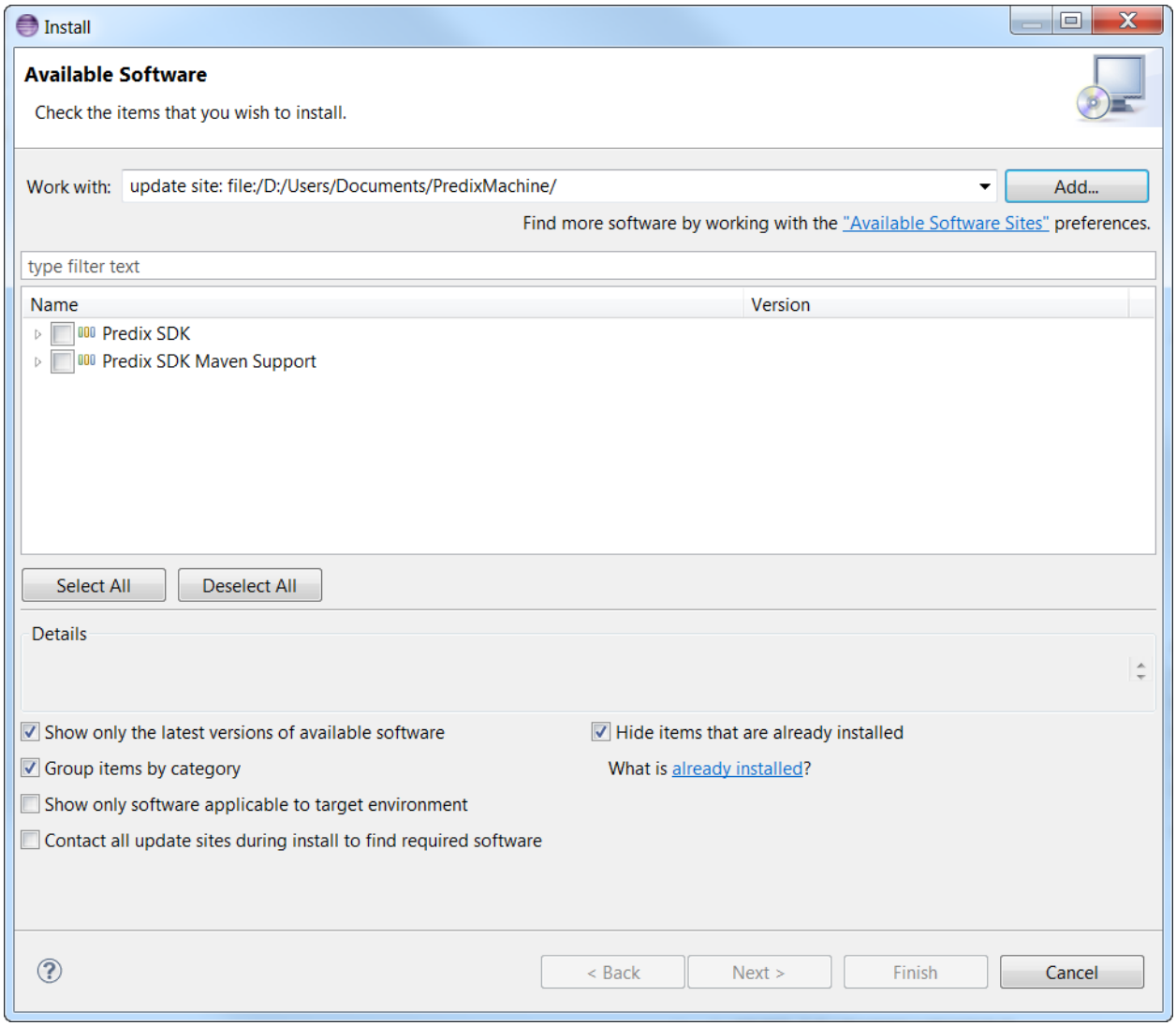
1. Open Eclipse.
   The **Welcome to Eclipse** page appears.
2. On the **Help** menu, select **Install New Software**.
   The **Available Software** window appears.

   Note: If you just downloaded the latest version of Eclipse, you should clear the **Contact all updates sites during install to find required software** checkbox to prevent Eclipse from searching for updates for all installed packages. If you have an older version of Eclipse, you can select the checkbox to perform this procedure, but it may take a long time.

3. Click the **Add** button.
   The **Add Repository** dialog box appears.



4. Click the **Local** button and then browse to the installation location where you unzipped the Predix Machine SDK files, select the **eclipse-plugins** folder, and click **OK**.
   The Predix Machine SDK and Predix Machine SDK Maven Support options now appear in the list of available software.

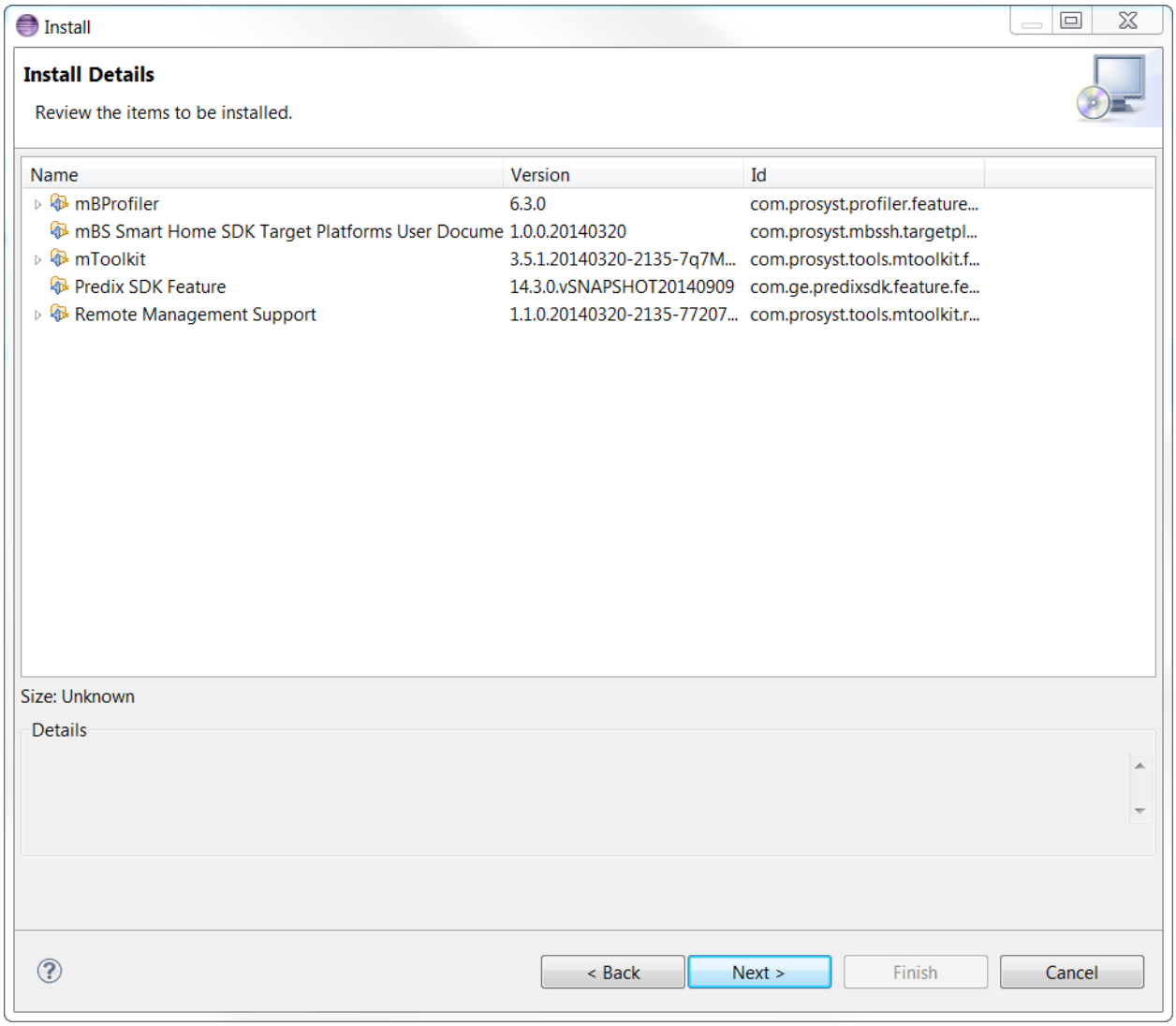


5. Select the following installation options:

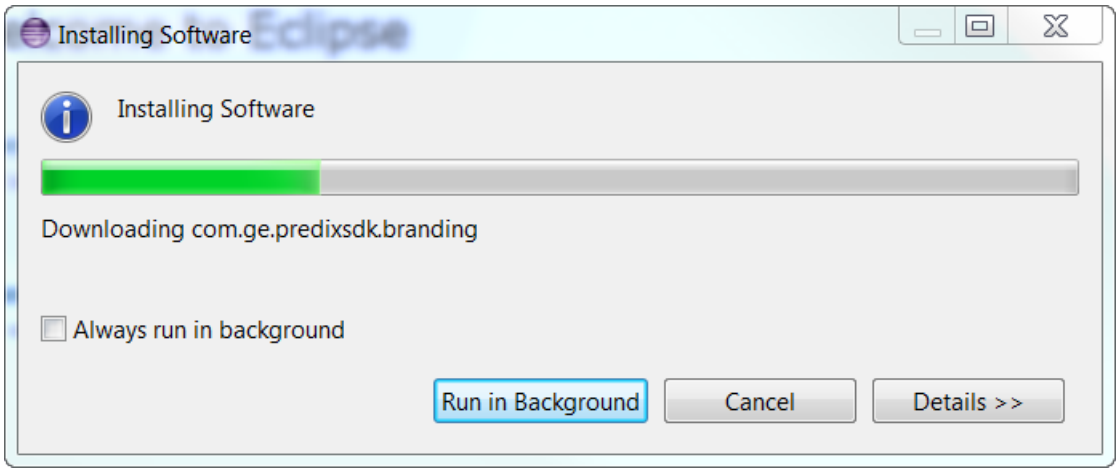
   - **Predix Machine SDK**
   - **Predix Machine SDK Maven Support**

   Note: Only select **Predix Machine SDK Maven Support** if you have installed the Maven Integration (m2eclipse) plugins. If M2E is not installed, the installation cannot continue until you install M2E or the Eclipse IDE for Java JEE Developers. This is available in the Eclipse Marketplace.

6. In the **Details** section, select the options you want to customize your installation. You can use the Default selections. Click the **Next** button.
   The **Install Details** window appears.

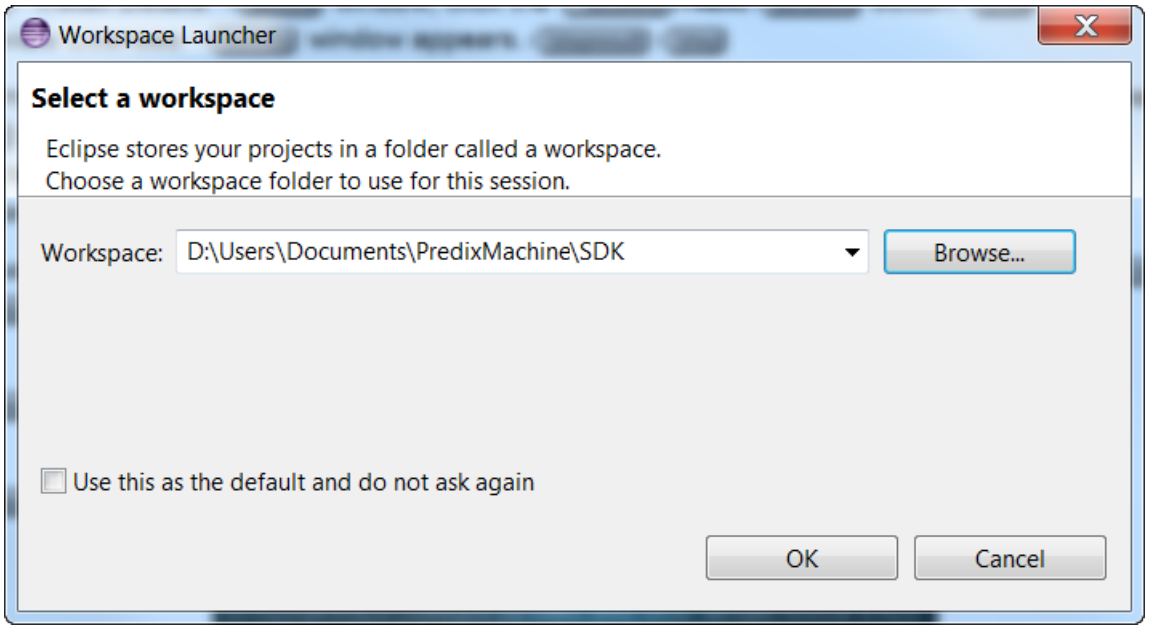


7. On the **Install Details** window, click the **Next** button.
   The **Review Licenses** window appears.
8. Review the terms of the license agreements then choose the **I accept the terms of the license agreements** option and click the **Finish** button.
   The **Installing Software** progress bar appears.



Note: If a Security Warning appears, click **OK** to continue.

9. When the software installation is complete, click the **Yes** button to restart Eclipse.
If you have not created a workspace for Eclipse, the **Select a Workspace** dialog box opens.

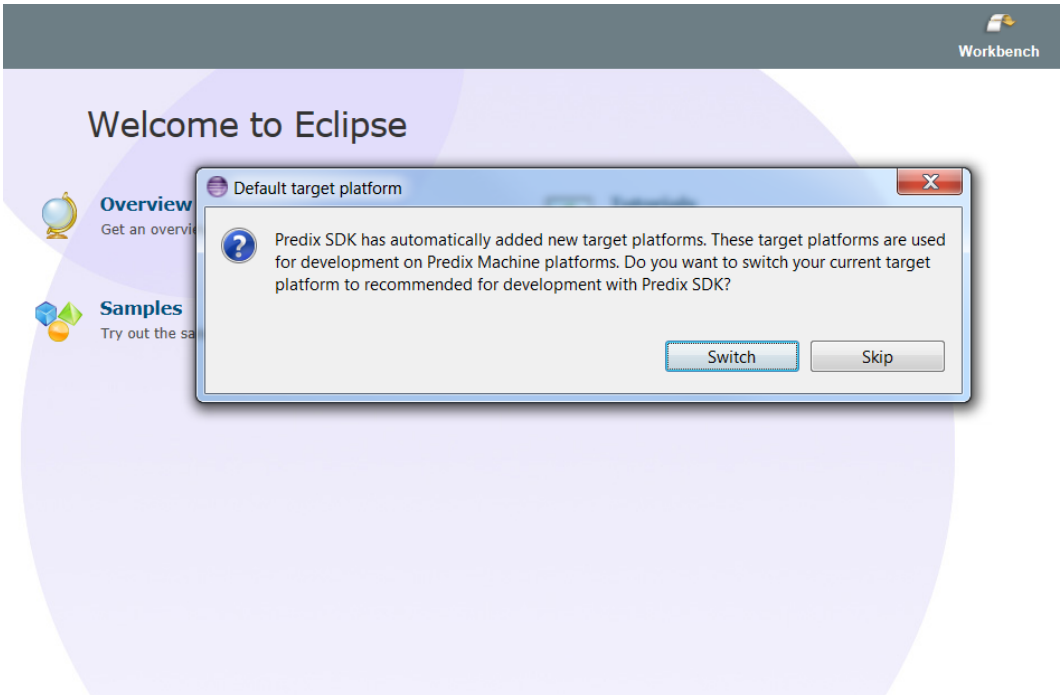


10. If you have not created a workspace, select the **Workspace** location and click **OK**.

Starting Eclipse

When you start Eclipse after installing the Predix Machine SDK, Eclipse configures your environment to use the SDK. During start-up, you are prompted to switch the Eclipse Plug-in Development Environment (PDE). The default PDE configuration uses OSGi Runtime components. If you do not switch, you can switch later.

1. Start Eclipse.
   The **Default target platform** dialog box appears.



2. Click **Switch**.

Sample Applications

The sample applications illustrate how to use Predix features.

Two sets of sample applications are provided: one for the Predix Machine container and one for cloud applications.

To view these samples, navigate to: `<SDK installation location>/samples` and extract the files from `sample-apps.zip` or `sample-cloud-apps.zip`.

The `sample-apps.zip` file includes:

- `sample-basicmachineadapter`
- `sample-configuration`
- `sample-container`
- `sample-custompolling`
- `sample-gitrepository`
- `sample-healthmachineadapter`
- `sample-hoover`
- `sample-httpclient`
- `sample-httpriver`
- `sample-mqttclient`
- `sample-mqttmachineadapter`
- `sample-security`
- `sample-storeforwardclient`
- `sample-subscriptionmachineadapter`
- `sample-websocketclient`
- `sample-websocketriver`
- `sample-websocketserver`

Note: See *Building Samples from a Command Line* on page 18 and *Running Samples in Eclipse* on page 20 or *Building Samples Using Eclipse IDE* on page 20.

The `sample-cloud-apps.zip` file includes:

- `httpdata`
- `httptunnel-server`

Note: Read the associated `readme.txt` files each cloud sample for instructions on how to use the samples.

Building Samples from a Command Line

1. Generate an API key.
   a) Log in to *https://artifactory.predix.io*.
   b) Click on your user name.
   c) Enter your password.
   d) Click **Unlock** to populate the API Key field.
   e) Copy the key.

2. In the `<user directory>.m2/settings.xml` file:
   a) Configure the proxy settings for Maven based on your own site needs. Ask your network administrator if you have questions about your requirements.
   b) Add a server entry with the following information. (You will use the API key that you copied in Step 1.)

```
<server>
      <id>artifactory.external</id>
      <username>{your predix cloud login}</username>
      <password>{encrypted password - API key}</password>
</server>
```

Note: If you are a GE employee on a GE network, do not include artifactory.external, see Step D.

   c) You may also have to set proxy settings for the HTTPS protocol.

```
<proxy>
   <id>optional</id>
   <active>true</active>
   <protocol>http</protocol>
   <username>proxyuser</username>
   <password>proxypass</password>
   <host>proxy.host.com</host>
   <port>80</port>
   <nonProxyHosts>*.host.com|localhost</nonProxyHosts>
 </proxy>
```

   d) If you are a GE employee, add the following server and repository entries.

```
<server>
    <id>artifactory.repo</id>
    <username>{sso}</username>
    <password>{encrypted password}</password>
</server>
```

```
<repository>
   <id>artifactory.repo</id>
   <name>artifactory.repo or name of your choosing</name>
   <url>https://devcloud.swcoe.ge.com/artifactory/repo</url>
</repository>
<repository>
   <id>artifactory.external</id>
   <name>artifactory.external or name of your choosing.</name>
   <url>https://artifactory.predix.io/artifactory/repo</url>
</repository>
```

3. To build samples from the command line:
   a) Navigate to `<SDK installation location>/samples` and unzip either the `sample-apps.zip` or `sample-cloud-apps.zip` files.
   b) Navigate to the <sample> folder and run the following command:

```
mvn clean install
```

Note: Some samples require items to be pushed to artifactory. (MQTT Client and Cloud Samples, for example.) You must also push a 3rd-party JAR file into your local m2 to build. Refer to the `readme.txt` files for instructions.

4. To build individual samples:
   a) Navigate to `<SDK installation location>/sample/<sample-name>`.
   b) Run the following command:

```
mvn clean install
```

Building Samples Using Eclipse IDE

Use the Eclipse IDE in which you installed the Predix Machine SDK to build samples.

1. Generate an API key.
   a) Log in to *https://artifactory.predix.io*.
   b) Click on your user name.
   c) Enter your password.
   d) Click **Unlock** to populate the API Key field.
   e) Copy the key.
2. In the `<user directory>.m2/settings.xml` file:
   a) Configure the proxy settings for Maven based on your own site needs. Ask your network network administrator if you have questions about your requirements.
   b) Add server entry with this information. (You will use the API key that you copied in Step 1.)

```
<server>
      <id>artifactory.external</id>
      <username>predix cloud login</username>
      <password>{encrypted password - API key}</password>
</server>
```

3. Launch Eclipse and create your own workspace.
4. Import samples by selecting: **File** > **Import** > **Maven** > **Existing Maven Projects**. Click **Next**.
5. Browse and select `<SDK installation location>/samples/sample-cloud-apps/sample` or `<SDK installation location>/samples/sample-apps/sample` as the root directory.
6. Click **Finish** to import all samples into your workspace.
7. Select the sample root directory and then select **Run** > **Run As** > **Maven Install**.

Running Samples in Eclipse

Create a new container using Eclipse. See *Generating a Predix Machine Container Using Eclipse* on page 24.

You can run samples using the Predix Machine SDK.

1. Access Eclipse and open your Predix Machine image.
2. In the **Bundles** section, click the **Add** button.
   The **Add bundles** window appears.
3. In the **Bundle type** list, select **Predix Samples Group**.
   A list of each of the Predix samples appears.
4. Select the samples you want to run and click **OK**.
5. Click **Run**.

Running Samples from Generated Containers

You can run samples from a container that was generated using scripts.

Before running any samples from a container that was generated using scripts, make sure the project is built.

1. Copy the JAR file from `<SDK installation location>/samples/sample-apps/sample/<sample-name>` to `<Predix Machine installation location>/machine/bundles/.`
2. Copy any configuration files needed by the sample from the `<SDK installation location>/samples/sample-apps/sample/configuration/machine` directory to the `<Predix Machine container location>/configuration/machine` directory.
3. Modify the `solution.ini` file by adding a `<bundle>` tag for each sample. For example::

   ```
   <bundle>
       <name>com.ge.dspmicro.{sample-name}-{version}.jar</name>
   </bundle>
   ```

   You can also copy and paste the existing `solution.ini` file for all samples and modify that file. The `solution.ini` file is located in `<SDK installation location>/samples/sample-apps/sample/machine/bin/vms`.
4. To run the container:
   a) Navigate to <Predix Machine container location>/bin .
   b) Run the following command: `start_predixmachine.sh` (for Linux) or `start_predixmachine.bat` (for Windows).

Generating a Predix Machine Container using Scripts

- Download Predix Machine from *https://artifactory.predix.io/artifactory/PREDIX-EXT/predix-machine-package/predixmachinesdk/16.2.0/predixmachinesdk-16.2.0.zip*.

  Tip: For Windows, put your SDK download in a high-level directory on your computer. For example, avoid `c:\\<directory>\<directory>\<directory>\Predix Machine SDK` instead use something like `c:\\PredixMachineSDK`. If you nest the download within several directories, the path name may be too long when you try to run the container.

- Download Eclipse with PDE runtime plug-ins; for example Eclipse IDE for Java EE Developers. This download should remain in the .zip or tar.gz format.
- Ensure that you have Maven installed. On a command line interface, type `mvn -version`. Your version should be 3.1 or above.

Note: Some features are only included in the container if you build the container with the Predix Machine SDK in Eclipse. See *Services Available through the Predix Machine SDK Only*.

You can use a command line interface to generate the following types of containers:

- PROV: Predix Machine Provision (includes only the JAR bundles that support provisioning)
- DEBUG: Predix Machine Debug with Predix Machine Web Console
- TECH: A Technician Console image
- CUSTOM <image file full path>: A Predix Machine container using a custom image you created in Eclipse
- [not specified]: Predix Machine default container

Note: Each of these container types maps to the **Workspace Image** selection in the **Image Description** dialog box in the Predix SDK in Eclipse.

1. Open a terminal window.
2. In the command line, navigate to the `<SDK download location>/predixmachinesdk-16.2.0/utilities/containers` folder.
3. Run one of the following commands.
   a) For Windows:

   ```
   GenerateContainers.bat <full path and name of downloaded Eclipse.zip file>
   -<type of container>
   ```

   b) For UNIX and Linux:

   ```
   GenerateContainers <full path and name of downloaded Eclipse.tar.gz file>
   -<type of container>
   ```

   For example, in Windows:

   ```
   GenerateContainers.bat
   D:\users\16.2.0\SDK\eclipse-jee-luna-SR2-win32-x86-64.zip -PROV
   ```

   The script creates the Predix Machine container in the `<SDK download location>/predixmachinesdk-16.2.0/utilities` folder.

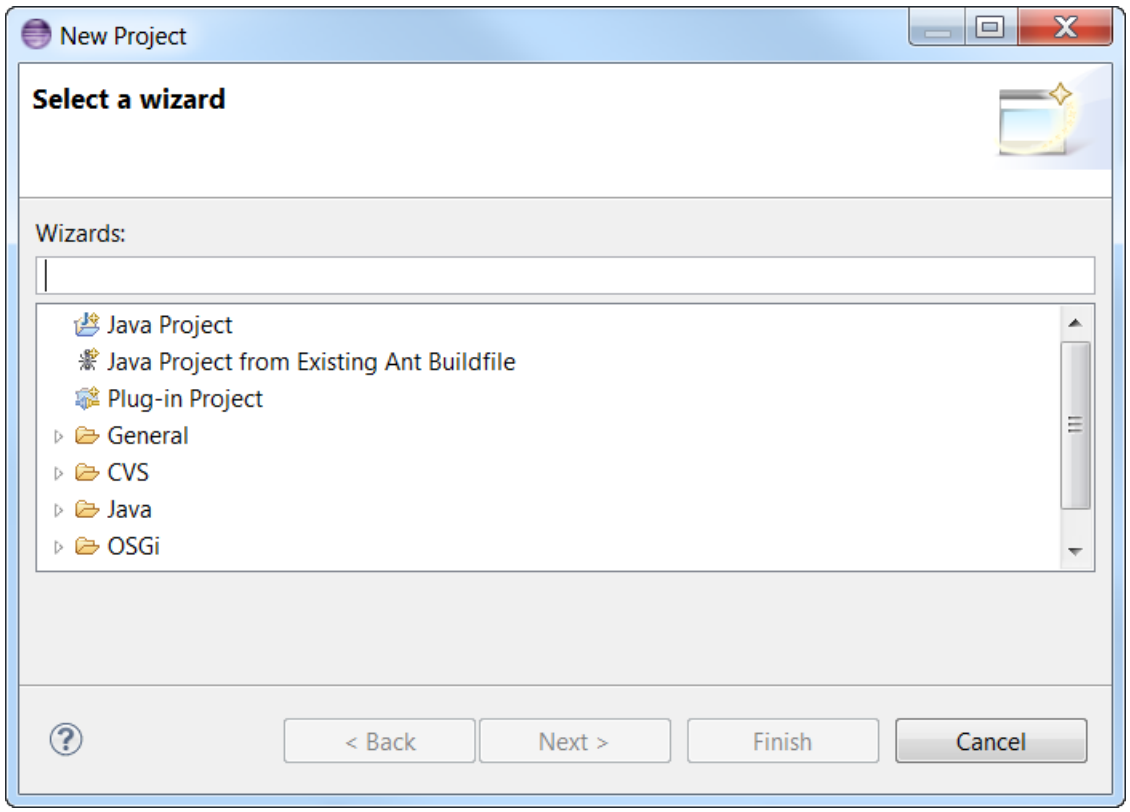
   Note: If you receive the following console error, you can ignore it: `java.lang.ClassCastException: org.eclipse.osgi.internal.framework.EquinoxConfiguration$1 cannot be cast to java.lang.String at`

```
org.eclipse.m2e.logback.configuration.LogHelper.logJavaProperties(LogHelper.java:26)
at org.eclipse.m2e.logback.configuration.LogPlugin.loadConfiguration(LogPlugin.java:189)
at org.eclipse.m2e.logback.configuration.LogPlugin.configureLogback(LogPlugin.java:144)
at org.eclipse.m2e.logback.configuration.LogPlugin.access$2(LogPlugin.java:107) at
org.eclipse.m2e.logback.configuration.LogPlugin$1.run(LogPlugin.java:62) at
java.util.TimerThread.mainLoop(Timer.java:555) at
java.util.TimerThread.run(Timer.java:505)
```

Generating a Predix Machine Container Using Eclipse

Follow these steps to generate a Predix Machine OSGi container using Eclipse.

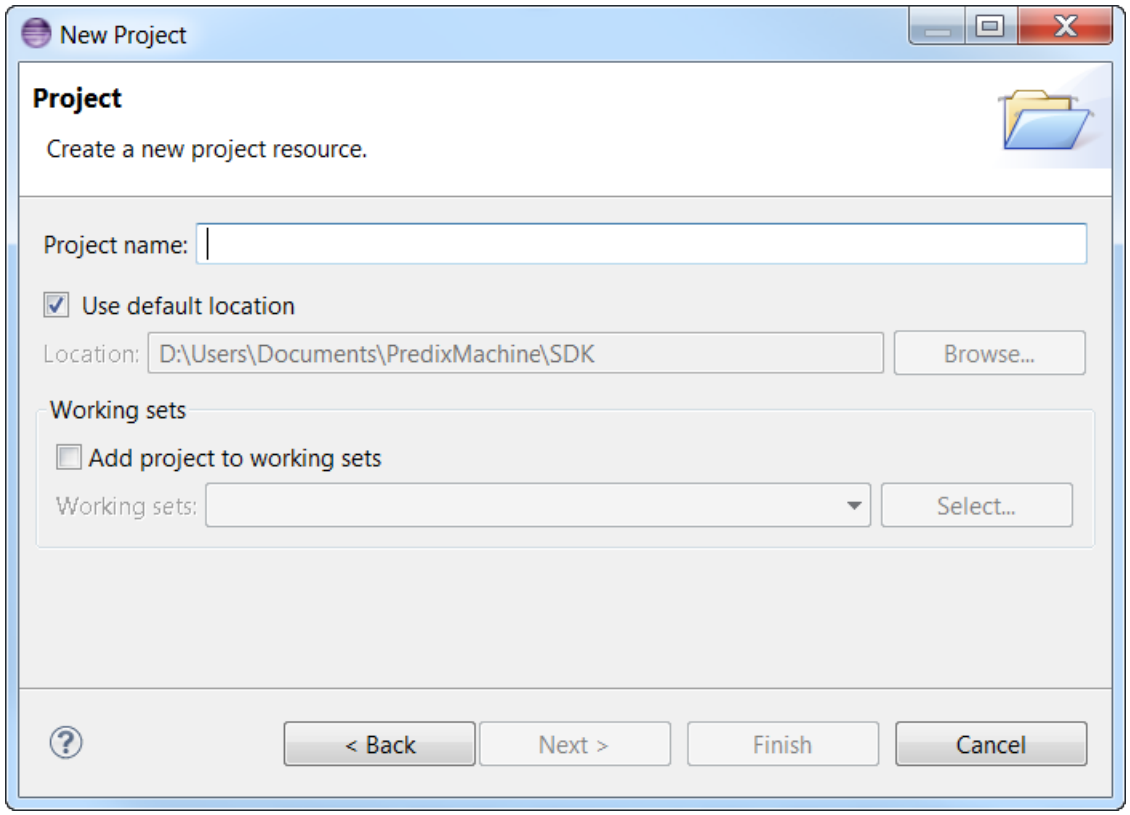
1. In Eclipse, select **File** > **New** > **Project** to create a project in which to create files and folders.
   a) In the **Navigator** panel, right-click and choose **New** > **Project**.
   The **New Project** box appears.



   b) Select the project wizard and click **Next**.

   Note: If the project contains only container images, select **General** > **Project**.

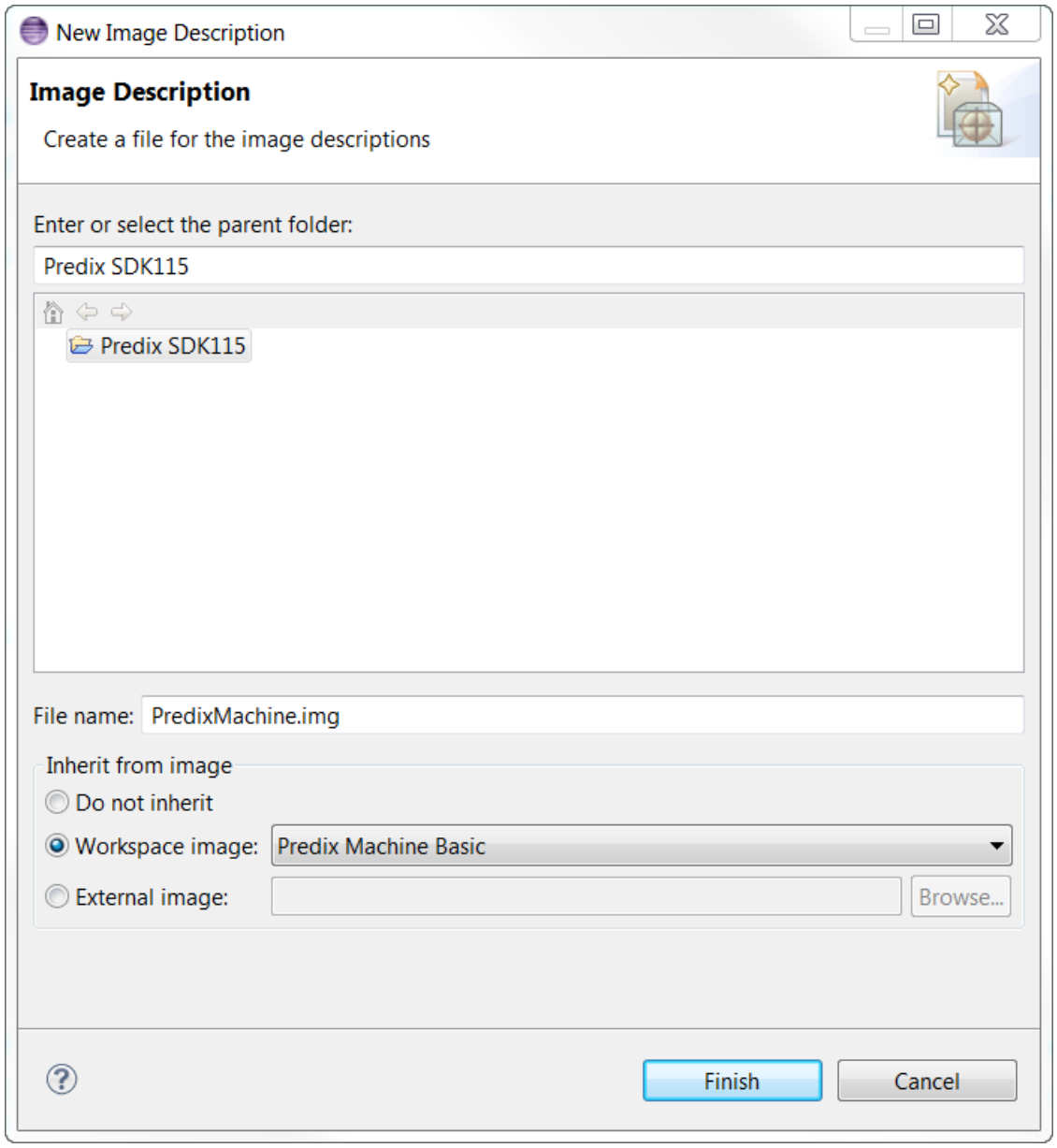
   The **Project** page appears in the **New Project** wizard.



c) In the **Project** page of the New Project wizard, assign a **Project name** to your project and click **Finish**. Your project appears in the **Navigator** pane.

2. Create a Predix Machine SDK image file.

a) Right-click in the **Navigator** panel, and choose **New** > **Image Description**. The **New Image Description** window appears.

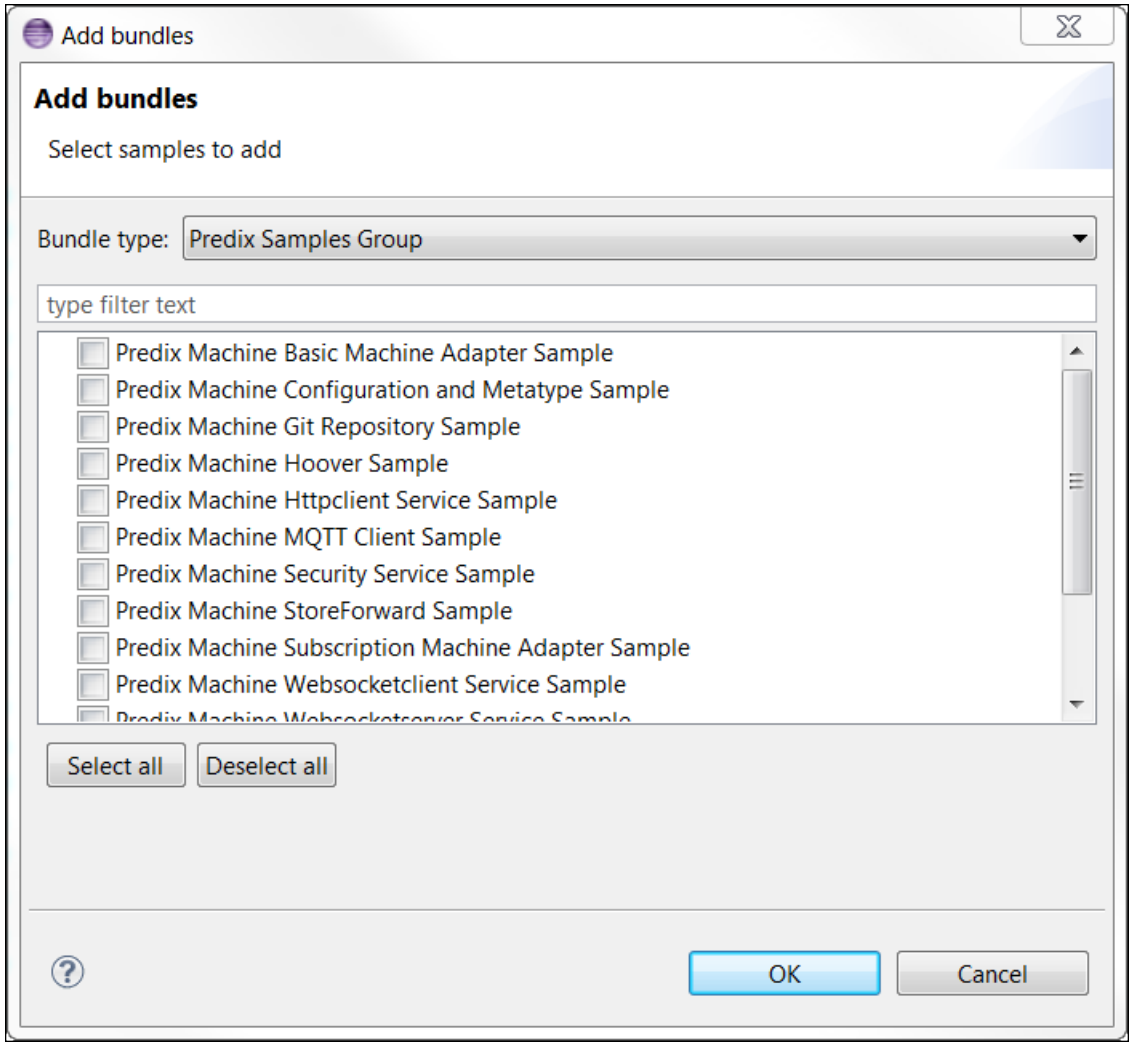


b) In the **File name** box, type a name for the file.
c) In the **Inherit from image** section, choose one of the following options for the **Workspace image** and click **Finish**.

- Predix Machine: Provides features in the containers
- Predix Machine Basic: Provides the container without any features
- Predix Machine Debug: Provides the Predix Machine Web Console
- Predix Machine Provision: Provides provisioning Support and the Predix Machine Web Console
- Predix Machine Technician Console: Provides the Predix Machine Technician Console

The image is created.

3. Add bundles to the container you will build.
   a) In the **Bundles** section, click the **Add** button.
   The **Add bundles** window appears.
   b) In the **Bundle type** list, select **Predix Features Group** to add bundles necessary for the various features.
   A list of each of the Predix features appears.



   See *Predix Machine SDK Overview* on page 6 to see the list of Predix Machine bundles that are part of each Predix Machine feature.

   Note: The Predix Machine SDK will prompt you to update image editor files when you open the image files. During the update, it will update older versions to current versions.

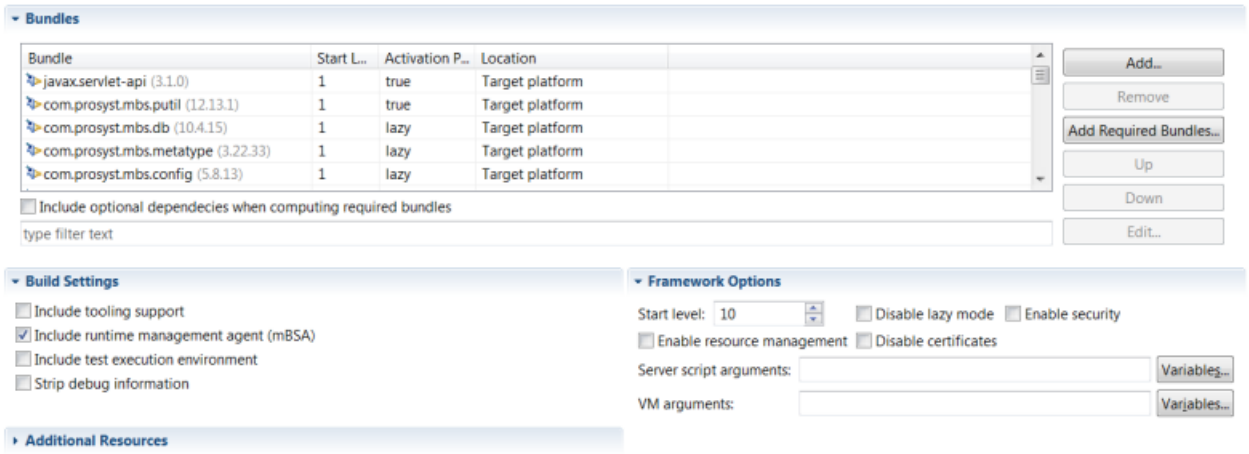
   c) Select the features to include in the container and click **OK**.

   Note: If you are adding a Maven bundle to the image, Eclipse will create an absolute path for the image. Edit the `pom` change the absolute path to a relative path, like. For example:

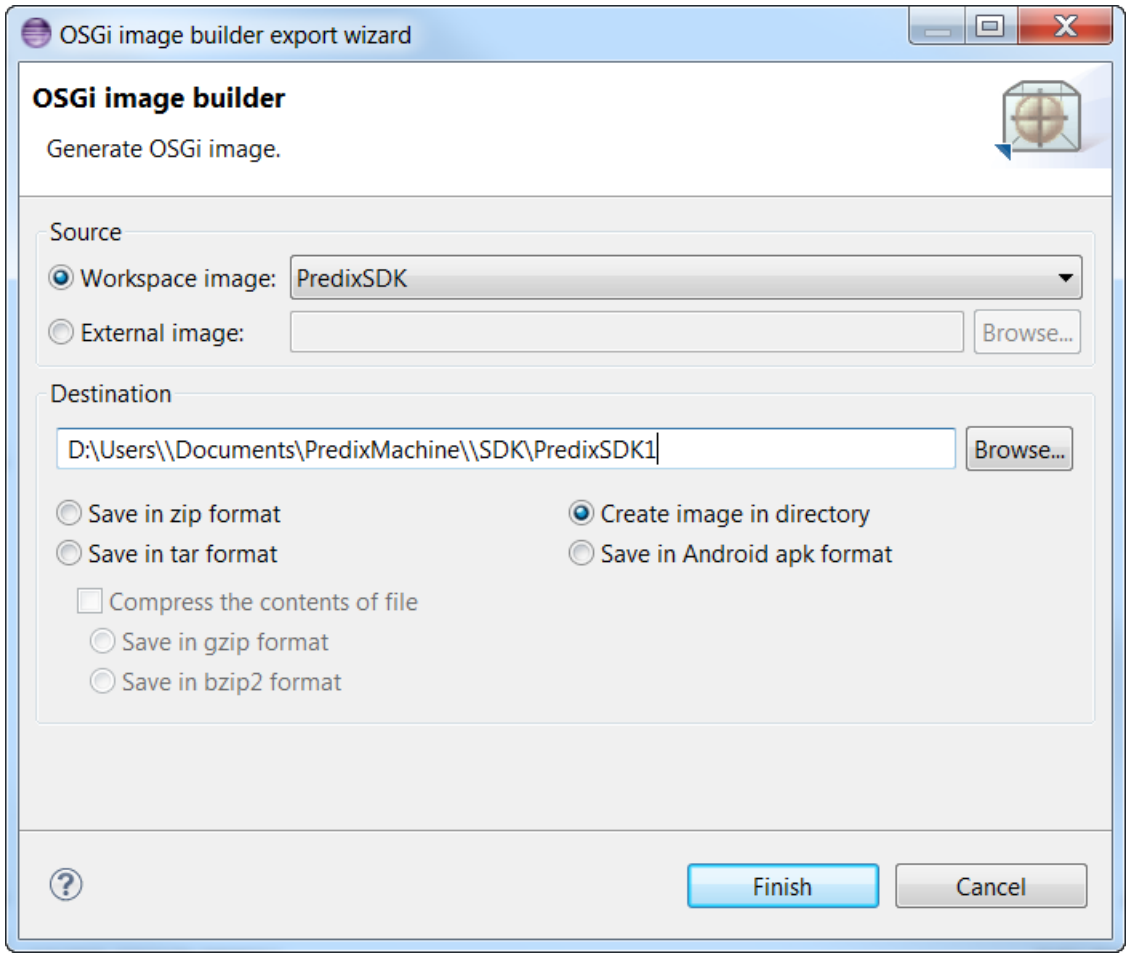
```
<element type="com.prosyst.tools.builder.osgiBundle"
location=maven:rel:../../../../gut/PredixMachine/master/machine-ge/ge-httpclient/po
```

4. To include the mBSA runtime management agent, which is used as a watchdog native application, in the **Build Settings** section, select the **Include runtime management agent (mBSA)** check box.

   Important: mBSA is required for supporting updates from the cloud.



5. Generate the container.
   a) In the **Testing** section, click the **Export** image link.
      The **OSGI image builder export wizard** appears.



b) In the **Workspace image** list, select the image you created in step 2.

c) In the **Destination** box, browse to the target location for the container and click **Finish**.
The wizard builds the container and stores it in the target location.

Note: You can also use the **Run**, **Debug**, and **Profile** options instead of **Export**. These enable you to test services that you may want to add.

Profiling sessions are limited to between 5 and 10 minutes.

When using Eclipse on Linux, the profiler does not work correctly unless `LD_LIBRARY_PATH` is set before launching Eclipse. This must be set to include `<workspace root>/.metadata/.plugins/com.prosyst.tools.mbsemulator/images/<image name>/osgi/lib/mbprofiler-agent/runtimes/linux-x86_64-generic` where <workspace root> is the location of the Eclipse workspace in use and <image name> is the name of the image file you created.

6. Start the generated container.

a) On your computer, navigate to the `<exported location>//bin/` folder.

b) Double-click the `start_predixmachine.bat` file for Windows or `start_predixmachine` for Linux/Mac, or run the bash script `start_predixmachine`.

Reference

Related Documentation

For more documentation related to Predix Machine, see "Edge Software and Services" at *https://www.predix.io/docs*.

Support

For issues related to the Predix Machine SDK, or Predix Machine services, log a support ticket at *https://predix.io./support/*.

Ask questions, scan our forums and read knowledge base articles posted to our Community Forum page at *https://forum.predix.io*.

Providing Documentation Feedback

Do you have feedback for the technical documentation team? We want to hear from you.

To provide feedback, log a comment in the Documentation space in the **Community** > **Forum**.