



Software Development Kit

Informatica MDM - Product 360

Version: 8.1.1

1 Table of Contents

1	Table of Contents	2
2	SDK Package	3
3	Prerequisites	3
3.1	Database	3
3.2	Java Development Kit.....	3
3.3	Eclipse	3
4	Installation of the SDK Online Help.....	4
5	Installation of the SDK	4
5.1	Install target definition.....	6
5.1.1	SDK Feature.....	12
5.2	Install launches and configuration	14
5.2.1	Launch PIM Server.....	16
5.2.2	Launch PIM Client.....	17
5.2.3	Launch PIM Repository Editor.....	18
5.2.4	Launch PIM Database Setup.....	19
5.3	Install examples	20
5.3.1	Examples at a glance.....	20

The SDK provides documentation on available API's, best practice development infrastructure, ready to use examples for customizing and integration as well as ready to use JUnit test projects

2 SDK Package

The SDK zip package has the following structure

Folder	Description
eclipse	Third-party eclipse plugins and external development tools
example	Example projects (customizing template projects, testing examples, PDE build examples, export templates, web access, etc)
preferences	Eclipse IDE preferences
projects	Example Eclipse project with configuration files required to start HPM server and client in development environment
targets	HPM runtime, javadoc, development tools and frameworks
workflows	Workflow runtime and templates

3 Prerequisites

3.1 Database

An available database server with the corresponding Product 360 schemas needs to be installed and configured to be accessible from the development machine. Refer to the installation manual for details

3.2 Java Development Kit

A Java Development Kit Version 1.8 in 64bit must be available on the machine to run the eclipse distribution. Optionally the server and client can run on 1.8 32bit Java. Due to licensing limitations the SDK package is delivered without a JDK. However, usually it's enough to just use the most recent 1.8 JDK available.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

3.3 Eclipse

You need to have Eclipse as your IDE, others will not work. The minimum required version is the Eclipse IDE 4.4 (Luna), however, we recommend to use the "*Eclipse for RCP and RAP Developers*" edition. The versions 4.5.x are also supported.

Eclipse 4.5

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/mars/R/eclipse-rcp-mars-R-win32-x86_64.zip

Most recent Eclipse version

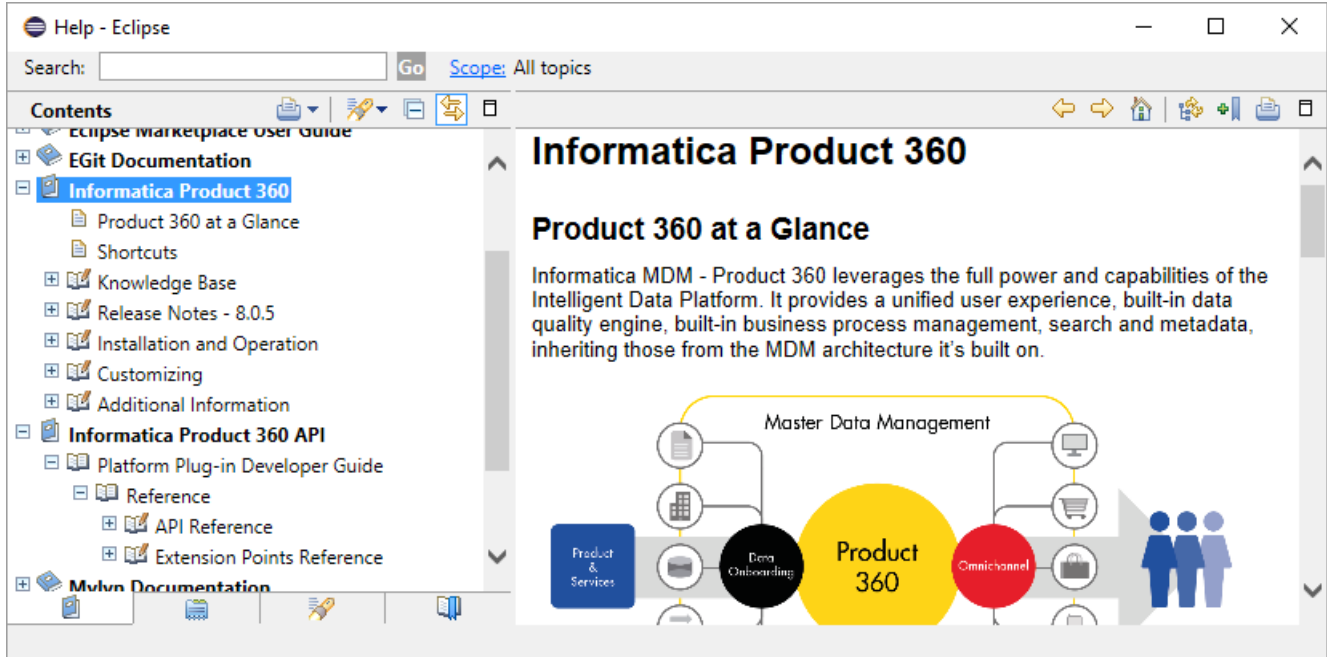
<http://www.eclipse.org/downloads/>

4 Installation of the SDK Online Help

The SDK online help contains the javadoc for the API, the list of available extension points as well as the general knowledge base and technical development articles. After installation it is accessible within Eclipse via Help -> Help Contents.


- Unzip the SDK distribution to an arbitrary place on your hard drive. In this example we use, C:\Informatica\PIM-SDK.
- Copy all plugins from the C:\Informatica\PIM-SDK\eclipse\sdkdoc folder to the dropins folder of your eclipse installation.
- Restart Eclipse

If you now open Help Contents inside of eclipse, you should see two sections named "Informatica Product 360" as shown on this picture.

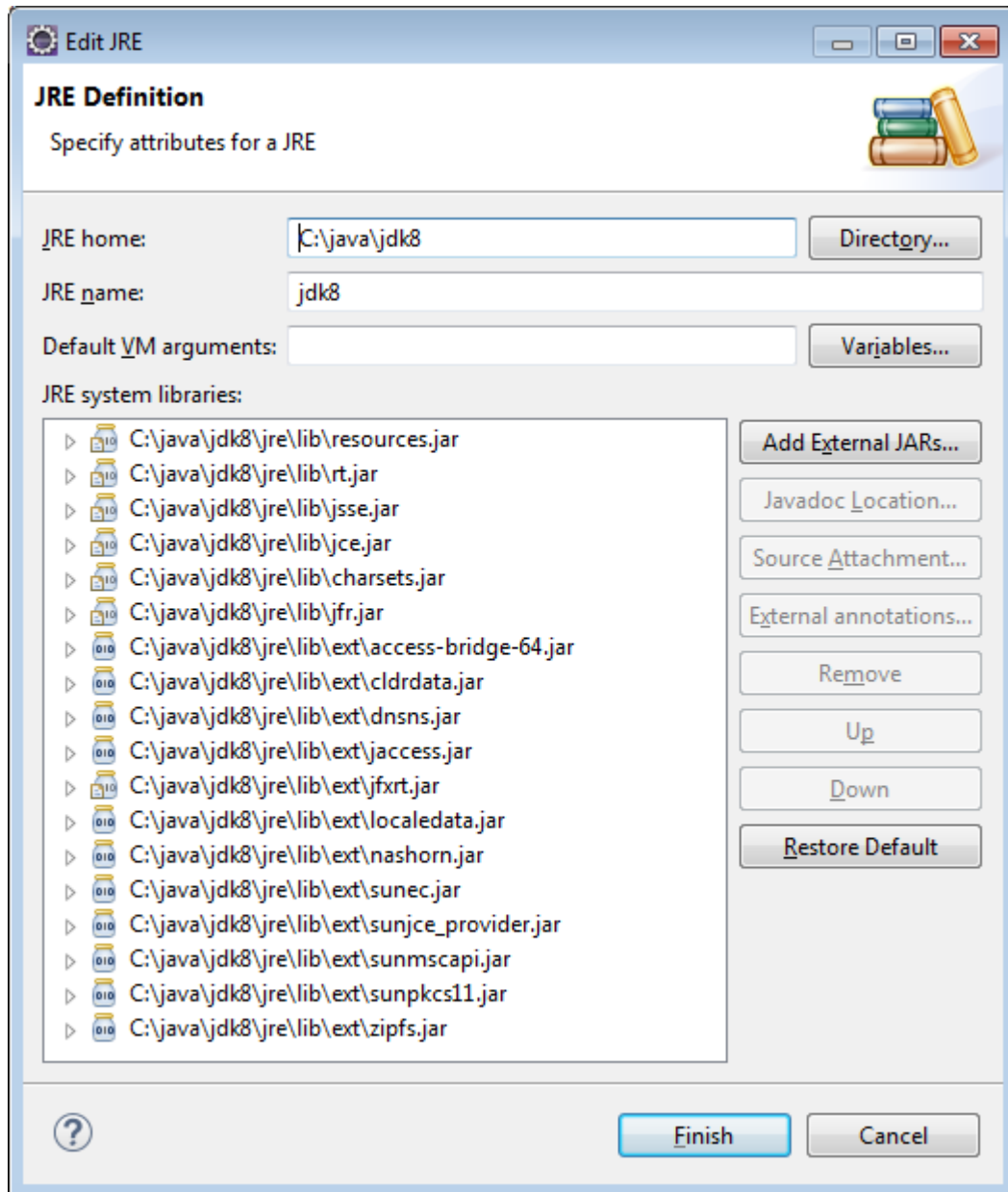


5 Installation of the SDK

- Run eclipse and choose an arbitrary workspace for the project you want to build. For convenience the SDK has a built-in project structure which you might find useful. We use C:\Informatica\PIM-SDK\projects\HENRI\workspace as the development workspace location for our "HENRI" project.
- (optional) Install 3rd party plugins from C:\Informatica\PIM-SDK\eclipse\3rdparty.
 - Unpack org.jbpm.gd.jpz into dropins folder of your eclipse installation.
 - org.jbpm.gd.jpz is a workflow development tool (you can also use latest versions of this plugin available on the internet).
 - We recommend to also install the Resource Bundle Editor plugin which makes it very easy to edit resource property files like the i18n files needed for multi-language support. This editor can be found on the Eclipse marketplace: <http://marketplace.eclipse.org/content/resourcebundle-editor>
- Go to **File > Import > Preferences** and import the Eclipse_4.4_Preferences.epf preferences into your workspace. This preference file is located in the C:\Informatica\PIM-SDK\preferences folder. Preferences contain Java formatter settings which satisfy the Informatica standard coding conventions.

 We suggest to use the provided development preferences, but you can also use your own conventions. In any case, we strongly recommend that the whole team uses the same conventions unless that you will have lots of "merge conflicts" with your source code management system.

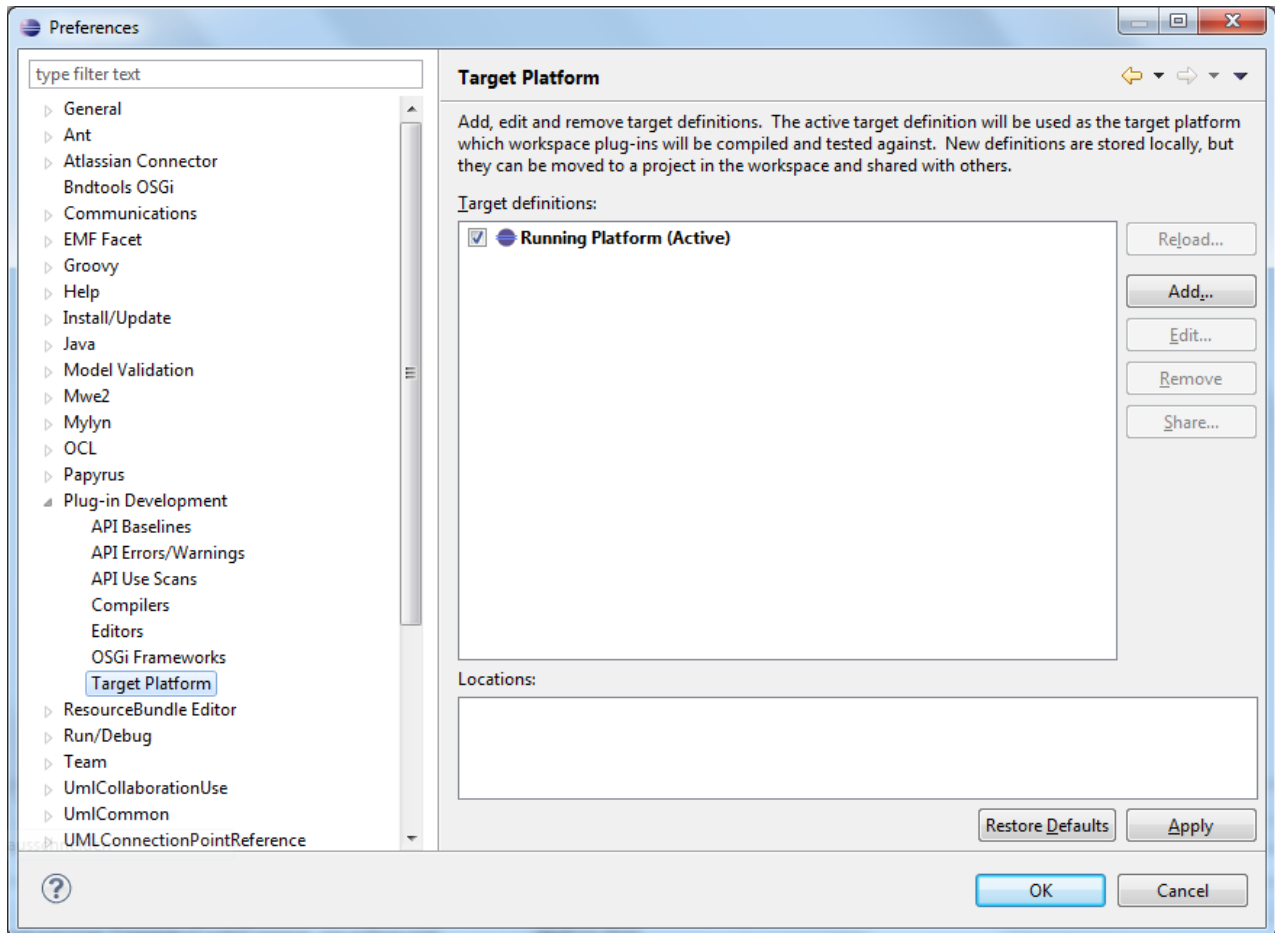
- Add the 1.8 JDK 64bit to the workspace, if not already done.
 - Open **Window > Preferences > Java > Installed JREs**
 - Add the JDK 1.8 to the list of available JREs and set the JRE instance name, for example jdk8.
- Make sure to use a JDK. By this means you have the Java source code available.*
- Set the installed JRE to "default" (check the checkbox).



- Check the Java compiler settings. Open **Window > Preferences > Java > Compiler** and make sure that the default compliance level is 1.8.

5.1 Install target definition

- Open **Window > Preferences > Plug-in Development > Target Platform**.



- Create new target definition: **Add > Start with empty target definition**. Set name to "PIM Target 64Bit"
- Open environment tab and set Target Environment.
 - *Operating System: win32, Windowing System: win32, Architecture: x86_64.*

Edit Target Definition

Target Content
Edit the name, description, and plug-ins contained in a target.

Name: PIM Target 64Bit

Locations | **Content** | Environment | Arguments | Implicit Dependencies

Target Environment
Specify the target environment. If left blank, the default environment variables from the host (running) platform will be used.

Operating System: win32

Windowing System: win32

Architecture: x86_64

Locale:

Java Runtime Environment
Specify the JRE or execution environment for this target. Selecting a named JRE will change the workspace default JRE setting.

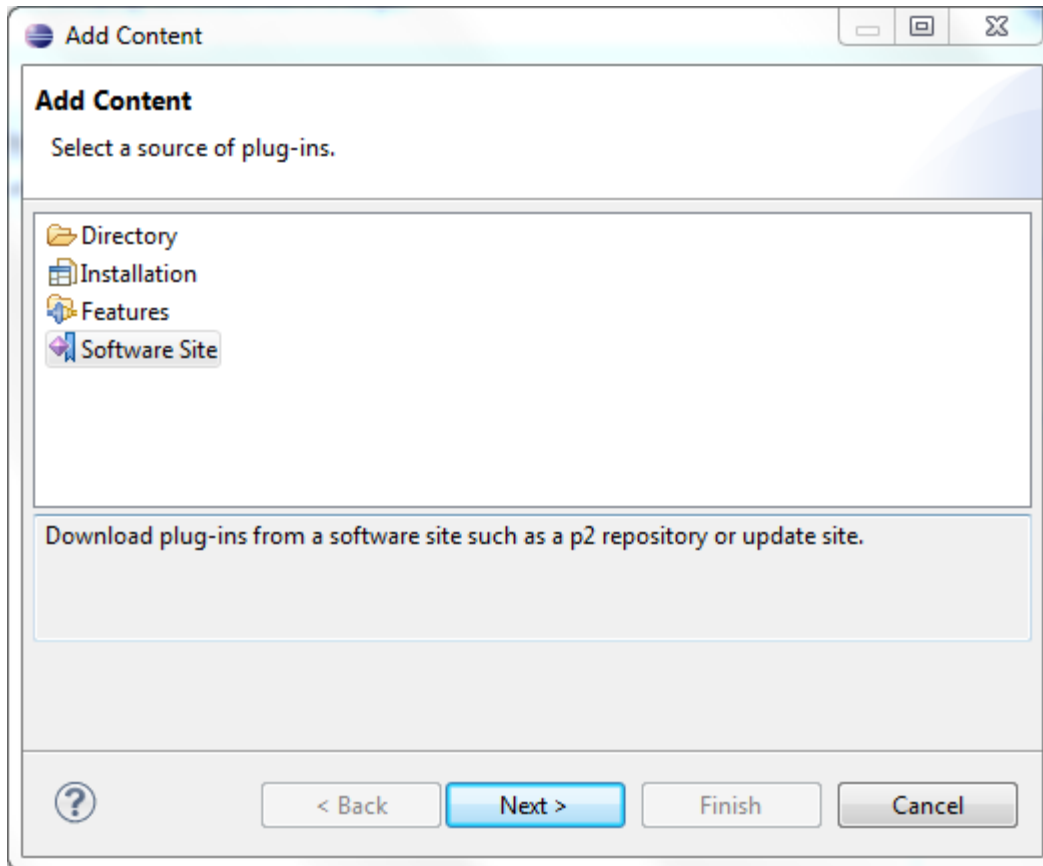
☒ Default JRE

☐ JRE name: jdk8

☐ Execution Environment: JavaSE-1.8

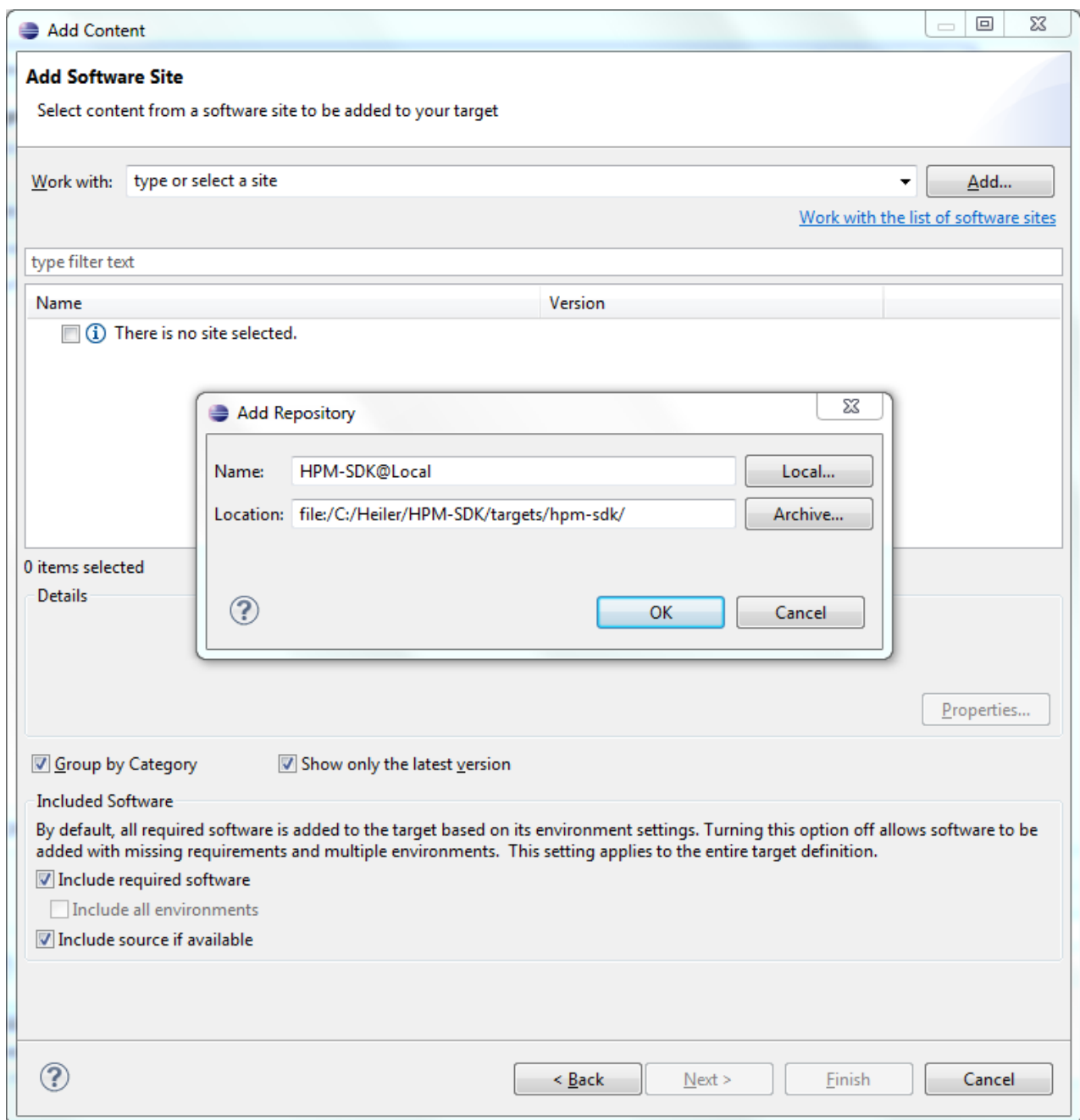
Finish Cancel

- Add new Software Site location: **Locations > Add > Software Site**

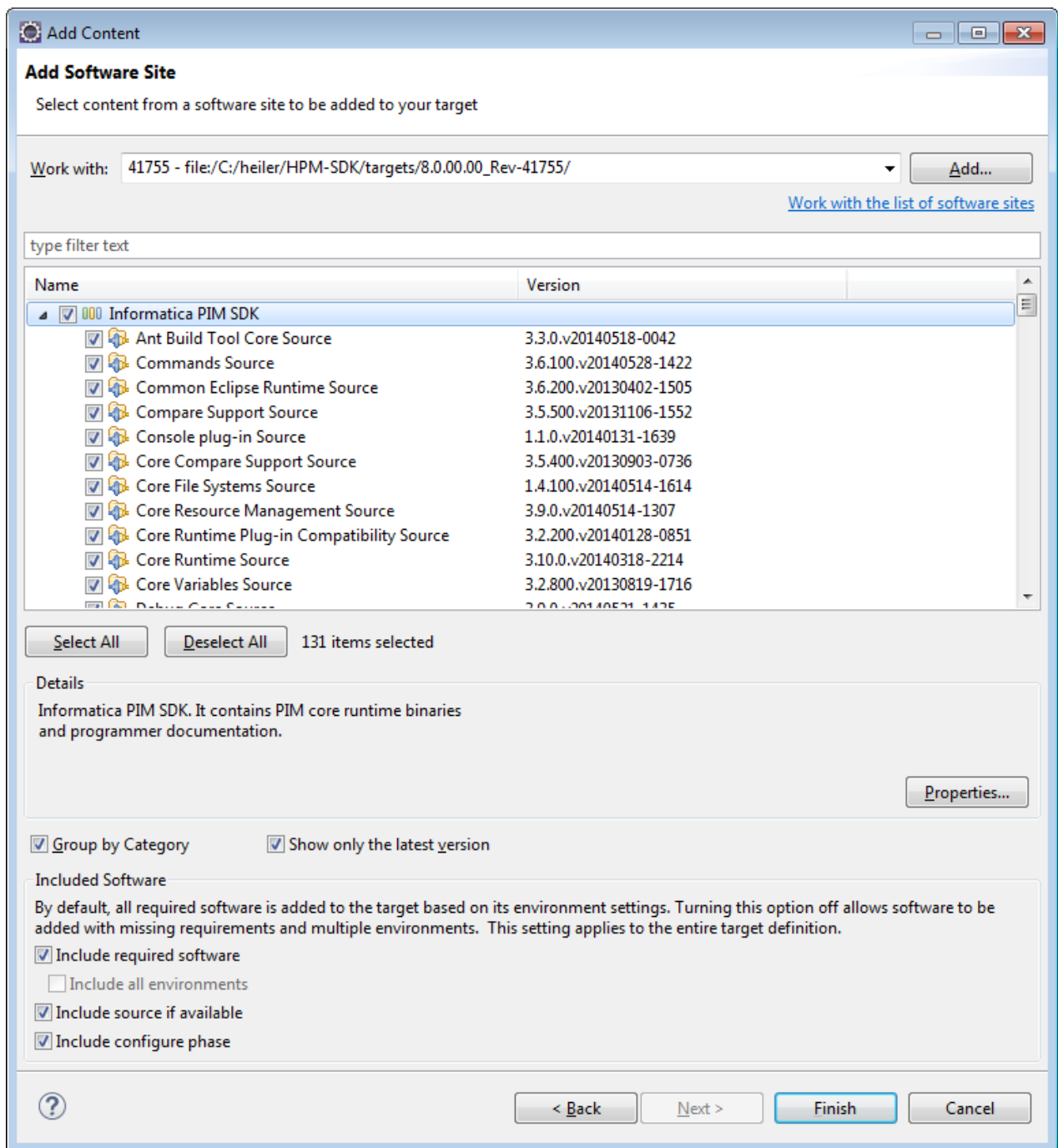


Add new local software site: **Add > Local...** Select

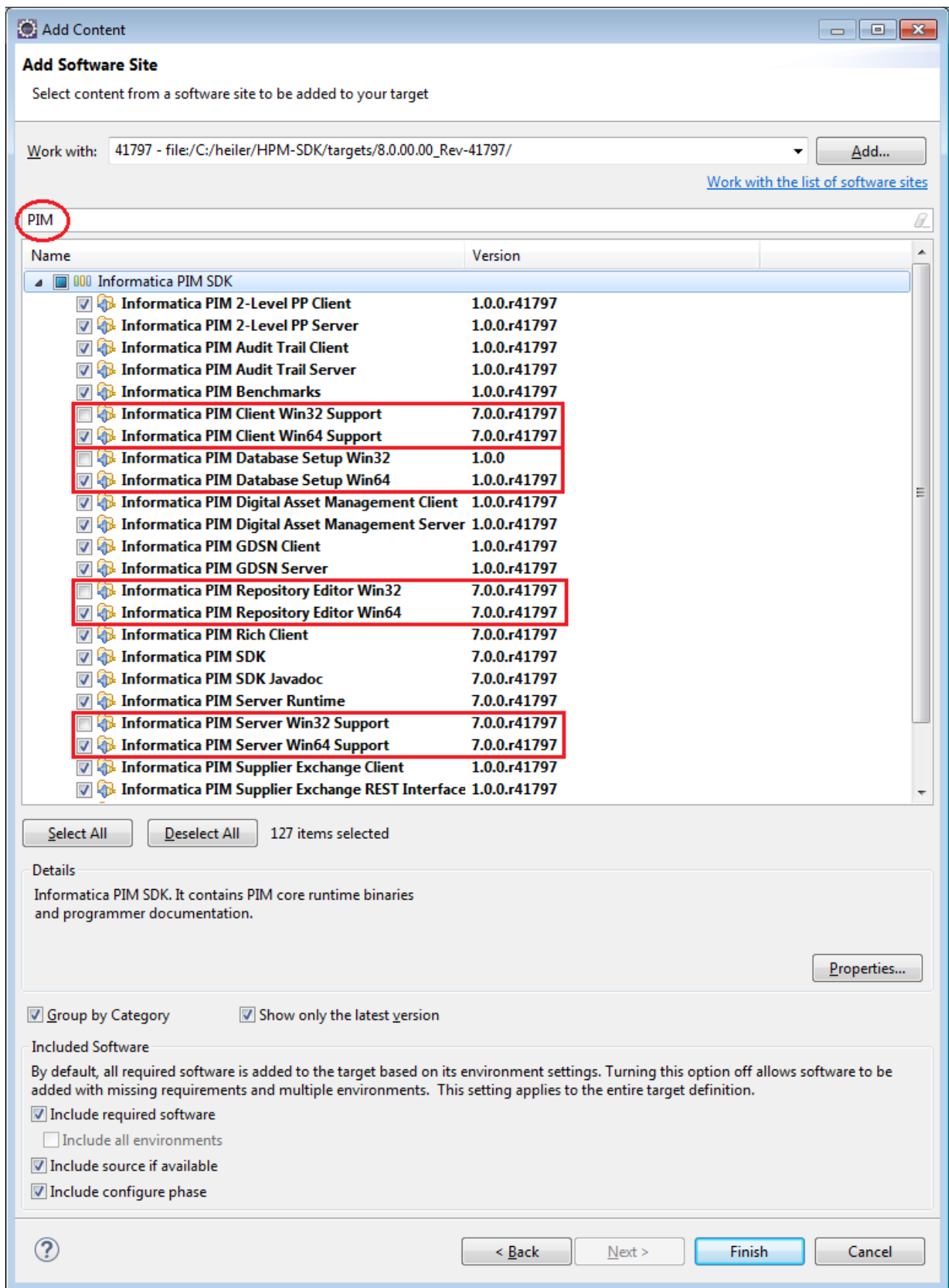
C:\Informatica\PIM-SDK\targets\8.0.x.xx\ folder and click **ok**.



- Now you will see a list of available features in the SDK including all 3rd Party and Eclipse features:



- Filtering only PIM features you can choose between 64bit and 32bit features:



5.1.1 SDK Feature

SDK Feature	Description
HPM Tools	PIM Runtime Tools (Command Inspector, Eclipse Plugin Spy and others)
Informatica PIM Benchmarks	Benchmark framework and standard benchmarks including sample data. Warning: this feature requires several hundreds of megabytes on hard disk.
Informatica PIM Rich Client	Rich Client runtime of PIM Desktop
Informatica PIM Client Win32 Support	Windows 32 platform client specific bundles
Informatica PIM Client Win64 Support	Windows 64 platform client specific bundles
Informatica PIM 2-Level PP Client	Rich Client runtime for the PIM Server for 2-tier Product Paradigm.
Informatica PIM Variant Client	Rich Client runtime for the PIM Server for 3-tier Product Paradigm.
Informatica PIM GDSN Client	Rich Client runtime for the GDSN Accelerator
Informatica PIM Audit Trail Client	Rich Client runtime for Audit Trail
Informatica PIM Repository Editor Win32	Repository editor (windows 32bit platform)
Informatica PIM Repository Editor Win64	Repository editor (windows 64bit platform)
Informatica PIM SDK	Container feature for SDK (contains all features listed here)
Informatica PIM SDK Javadoc	Contains the javadoc for the server and client binaries.
Informatica PIM Server Runtime	Server runtime
Informatica PIM Testing Framework	Testing framework including junit, mockito and eclipse testing framework
Informatica PIM Server Win32 Support	Windows 32 platform server specific bundles (GraphicsMagick, eclipse launcher, etc)
Informatica PIM Server Win64 Support	Windows 64 platform server specific bundles (GraphicsMagick, eclipse launcher, etc)

SDK Feature	Description
Informatica PIM 2-Level PP Server	Server runtime for the PIM Server for 2-tier Product Paradigm.
Informatica PIM Variant Server	Server runtime for the PIM Server for 3-tier Product Paradigm.
Informatica PIM GDSN Server	Server runtime for the GDSN Accelerator
Informatica PIM Audit Trail Server	Server runtime for Audit Trail
Informatica PIM Digital Asset Management Server	This feature contains the plugin(s) that are needed to use Digital Asset Management system
Informatica PIM Digital Asset Management Client	This feature contains the plugin(s) for the use of Digital Asset Management system specific actions in the rich client
Informatica PIM Supplier Exchange Server	This feature contains the plugin(s) for the use of PIM Supplier Portal system
Informatica PIM Supplier Exchange Client	This feature contains the plugin(s) for the use of PIM Supplier Portal system specific actions in the rich client
IBM Websphere Server	This feature contains the plugin(s) that are needed to use IBM Websphere integration
IBM Websphere Client	This feature contains the plugin(s) for the use of IBM Websphere specific actions in the PIM Desktop
Informatica PIM Web Client	Web Access components that are needed when starting the PIM Desktop (Web Action Rights)
Informatica PIM Web Server	Web Access runtime for the PIM Server.
Informatica PIM Web 2-Level PP Server	Web Access runtime for the PIM Server for 2-tier Product Paradigm.
Informatica PIM Web Variant Server	Web Access runtime for the PIM Server for 3-tier Product Paradigm.
Informatica PIM Web GDSN	Web Access runtime for the GDSN Accelerator

Select at least

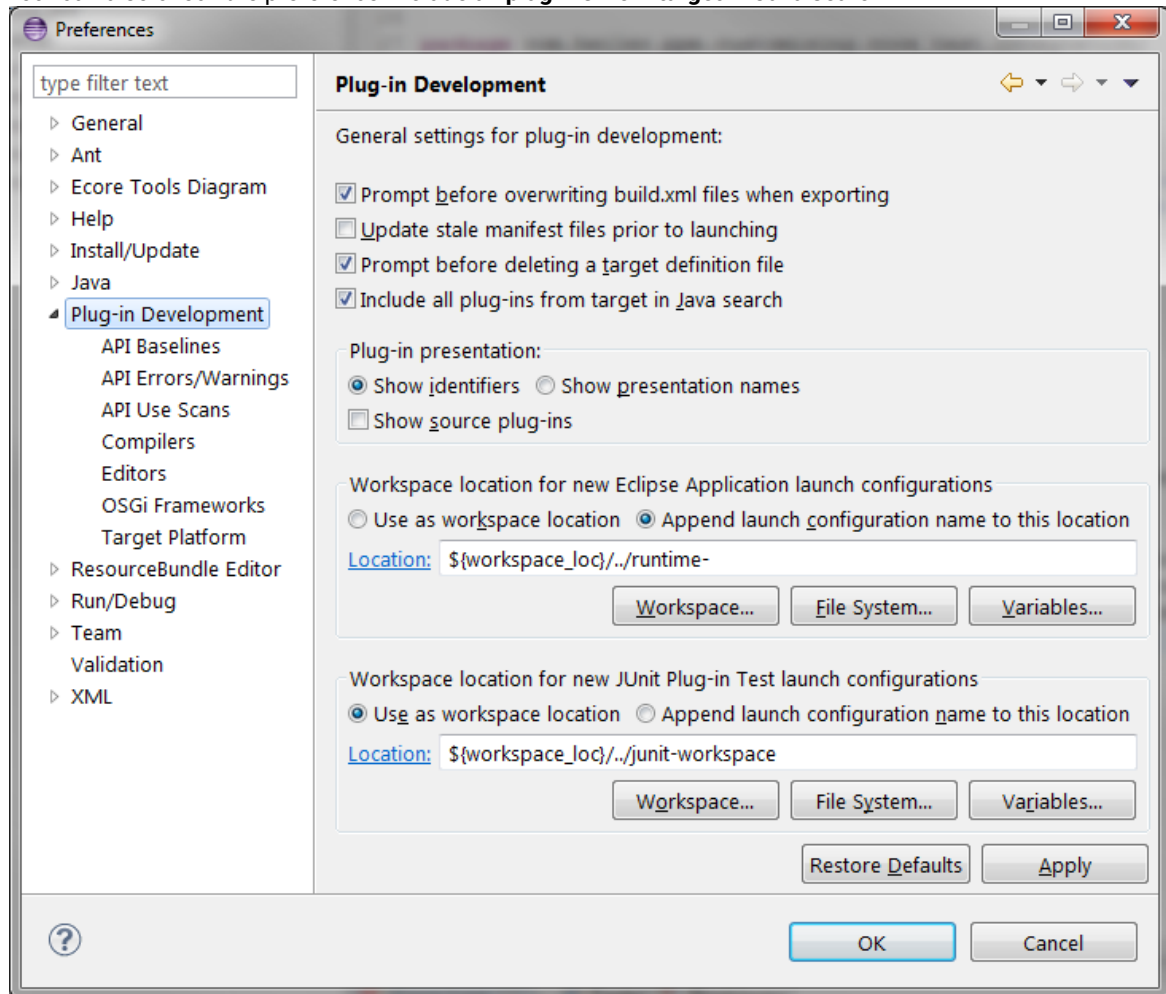
- *'Informatica PIM Rich Client'*,
- *'Informatica PIM Client Win64 Support'*,
- *'Informatica PIM Repository Editor Win64'*,
- *'Informatica PIM Server Win64 Support'*.

For full-fledged development you will need all features and respective platform fragments.



You can not import both Win32 and Win64 fragments unless you selected 'Include all environments' in the SDK software update site configuration. Note that the Product Manager SDK feature contain also the win64 fragments.

- Click **next**. Eclipse will now unpack target platform bundles and features into its internal folders. Click **finish** and select 'PIM Target 64Bit' as default target platform.
- Add target platform bundles to java search. **Window > Show View > Plug-in development > Plug-ins**.
 - In the Plug-ins view toolbar click on 'Add All Plug-ins to Java Search'.
 - You can also check the preference 'Include all plug-ins from target in Java search'



5.2 Install launches and configuration

- Import the launch and configuration projects into your workspace.
- Open **File > Import > Existing Projects into Workspace**.
- Select the C:\Informatica\PIM-SDK\projects\HENRI directory for the root directory.

- Check the configuration and the launches project and press **finish**

Import

Import Projects

⚠ Some projects cannot be imported because they already exist in the workspace

☒ Select root directory: D:\temp\PIM_8.0_SDK\projects\HENRI Browse...

☐ Select archive file: Browse...

Projects:

- ☒ configuration (D:\temp\PIM_8.0_SDK\projects\HENRI\configuration)
- ☒ launches (D:\temp\PIM_8.0_SDK\projects\HENRI\launches)

Select All
Deselect All
Refresh

Options

- ☐ Search for nested projects
- ☐ Copy projects into workspace
- ☒ Hide projects that already exist in the workspace

Working sets

- ☐ Add project to working sets

Working sets: Select...

? < Back Next > **Finish** Cancel

- Configure the configuration settings of your project. In the configuration project in your workspace you will find two sub folders, client and server. The client folder holds all configuration files for the client start, usually you do not need to modify anything here. The server folder contains all required configuration files for the server start. At least the `server.properties` file needs to be configured. In the server folder you will find several `server.properties` templates (one per supported database). There are a few properties that must be set in order to start the server: See the corresponding section in the PIM Core Installation Manual for details on this configuration.

Example: server properties

```
filestorage.dir.shared           = C:/Informatica/PIM-SDK/projects/HENRI/runtime
license.customer.file.local = ${filestorage.dir.shared}/license/<YourLicenseFile.license>
license.customer.key           = <CustomerKey>
db.default.server              = localhost
db.default.dir                 = C:/Informatica/PIM-SDK/projects/HENRI/runtime/database
db.default.schema.suffix      = _SDK805
```

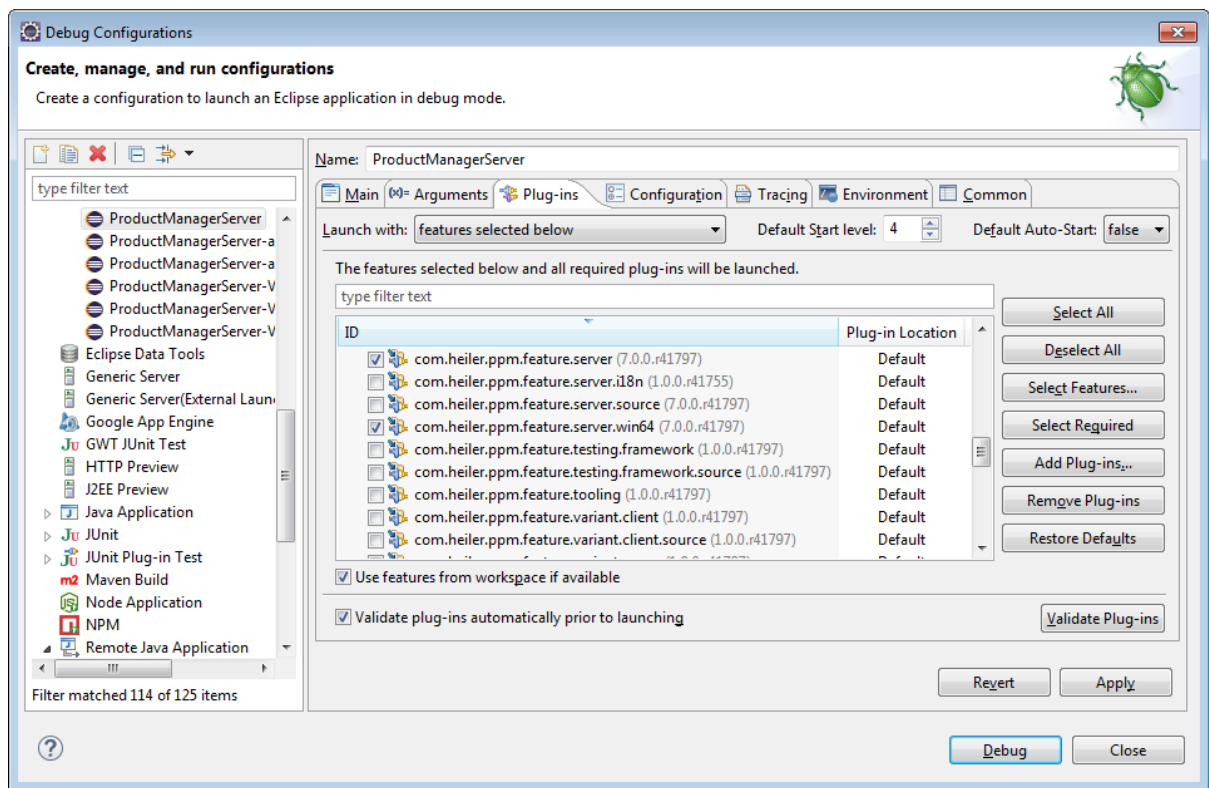
i Before setting up your SDK environment you can set up a standalone Product 360 runtime system for your project, you will need the development database anyway. After this you can use the configuration files from this working system and copy them to the configuration/client and configuration/server directory.

i For setting up your SDK environment you need a "Informatica PIM 8.0" license. Consulting and external implementation partners can request this license from the support department.

- Copy your license file to the directory C:/Informatica/PIM-SDK/projects/HENRI/runtime/license

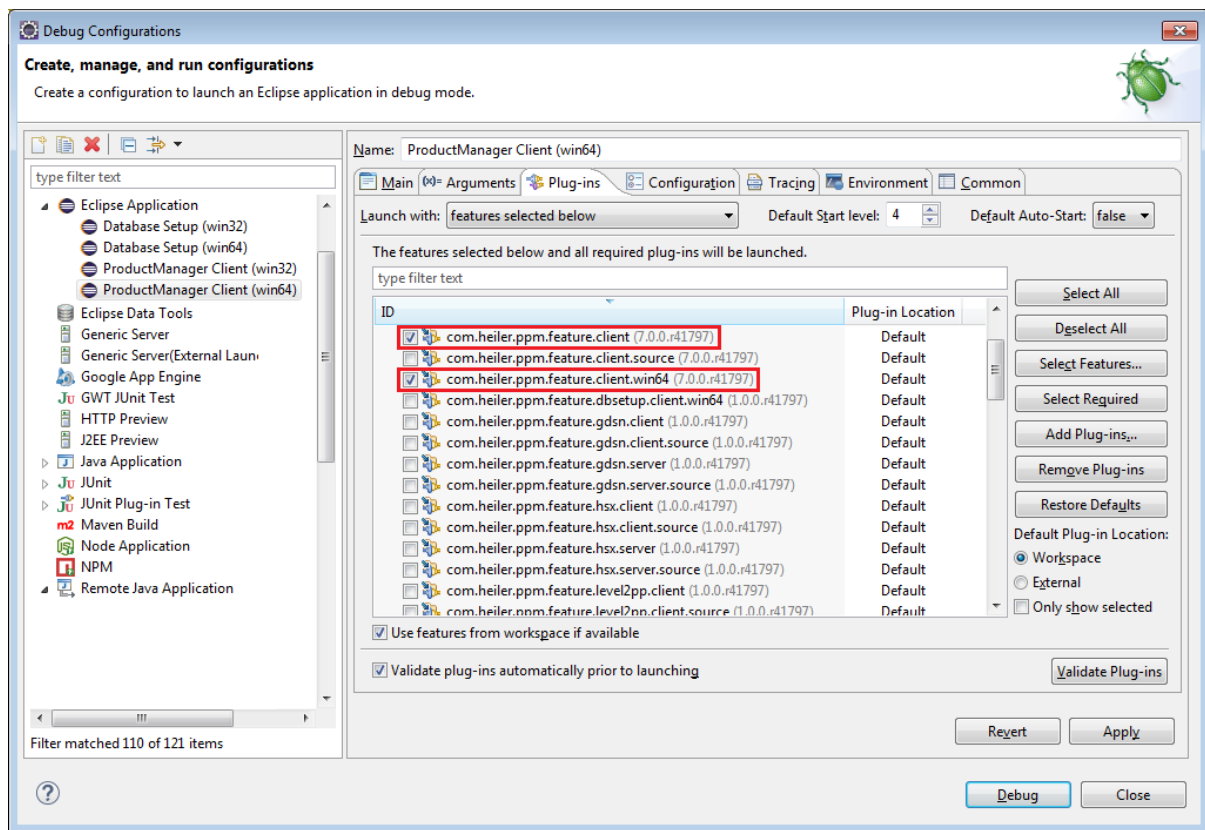
5.2.1 Launch PIM Server

- Go to **Run > Run Configurations > Eclipse Application > ProductManager Server**.
- Select '*com.heiler.ppm.platform.server.app*' application.
- Select Plug-ins tab and make sure that '**Launch with**' is set to '*features selected below*'.
- Check '*com.heiler.ppm.feature.server*' and '*com.heiler.ppm.feature.server.win64*'.
- (Optional) To run P360 with Informatica Media Manager, select '*com.heiler.ppm.feature.opasg.server*', too.
- (Optional) To use Web Access, select '*com.heiler.ppm.feature.web*', too.
- Add `-registryMultiLanguage` to the Program arguments
- Press run.



5.2.2 Launch PIM Client

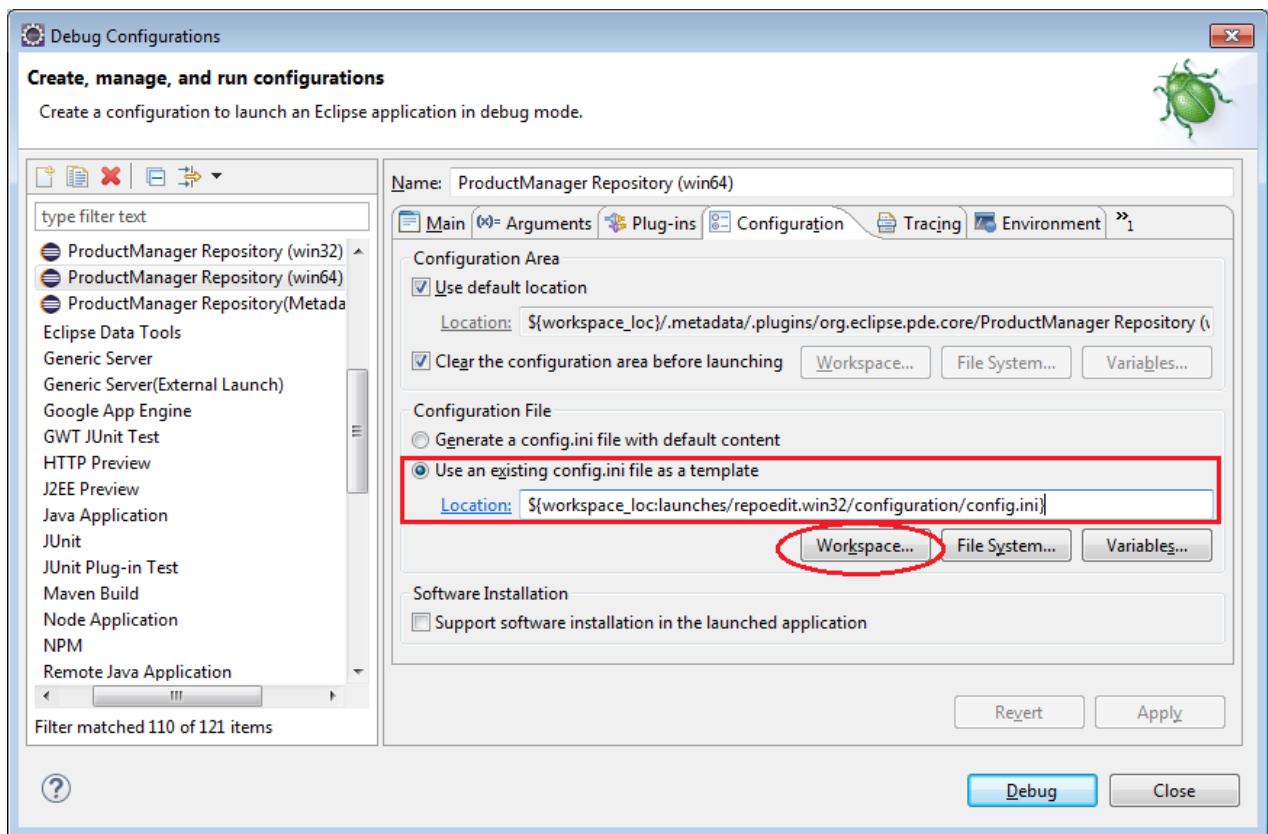
- Go to **Run > Run Configurations > Eclipse Application > ProductManager Client (win64)**.
- Select '*com.heiler.ppm.main.product*' product.
- Make sure runtime JRE is set to a 64bit JRE or JDK.
- Select Plug-ins tab and make sure that '**Launch with**' is set to '*features selected below*'.
- Check '*com.heiler.ppm.feature.client*' and '*com.heiler.ppm.feature.client.win64*'.
- (Optional) To use Web Access, select '*com.heiler.ppm.feature.web.core*', too (needed to configure Web Permissions).
- Add `-registryMultiLanguage` to the Program arguments
- Press run.



- Enter default credentials in the authentication dialog (Username: Administrator, Password: Administrator).
- Launch Web Access
 - Open your browser and enter the following url: `http://localhost:1512/pim` (replace *localhost* with the *dest.host* from *server.properties* if configured)
- Congratulations, you're ready to start developing with Informatica PIM!

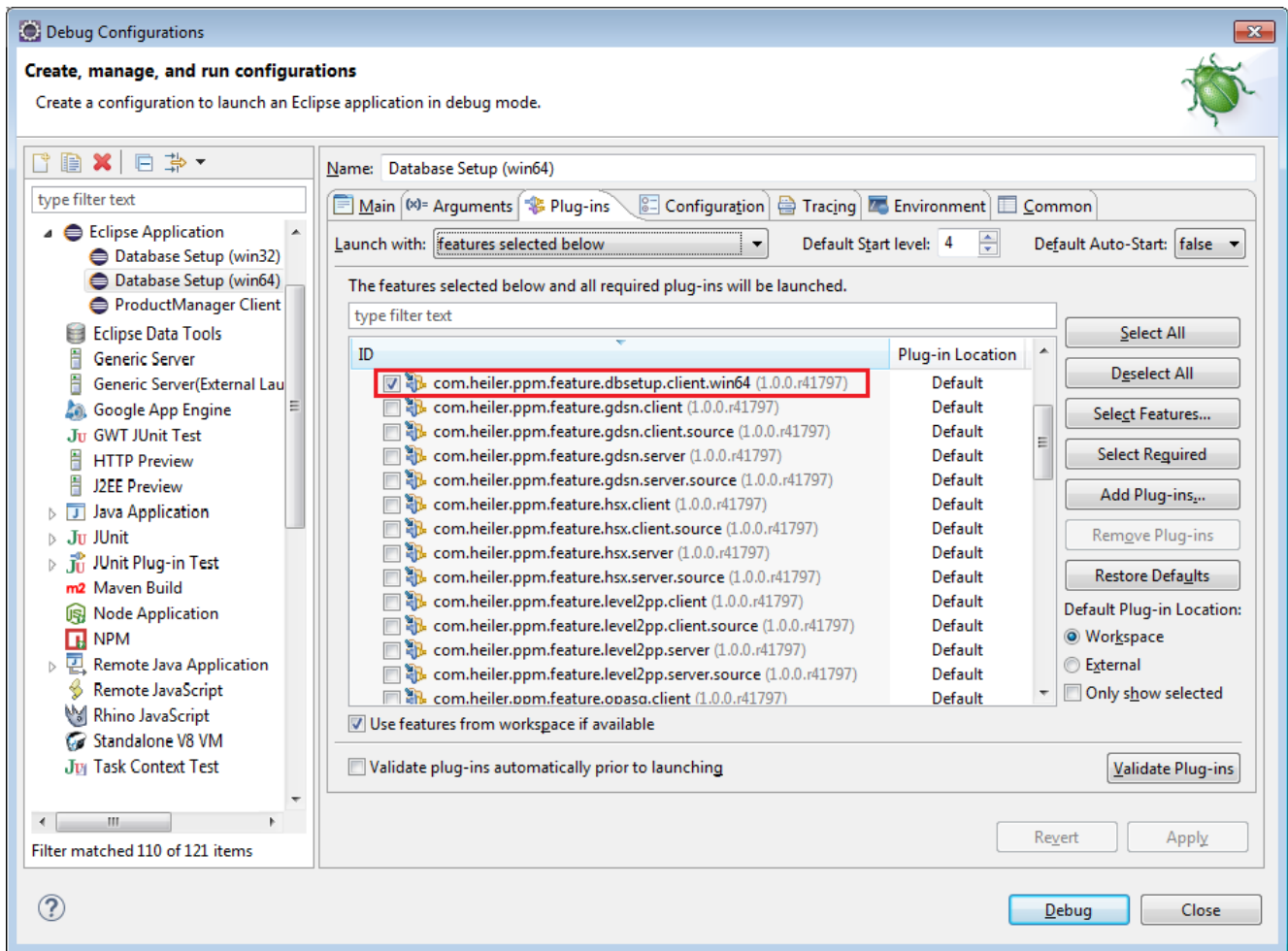
5.2.3 Launch PIM Repository Editor

- Go to **Run > Run Configurations > Eclipse Application > ProductManager Repository (win64)**.
- Select '*com.heiler.ppm.repository.core.editor.product*' product.
- Make sure runtime JRE is set to a 64bit JRE or JDK.
- Select Plug-ins tab and make sure that '**Launch with**' is set to '*features selected below*'.
- Check '*com.heiler.ppm.feature.repository.win64*'.
- Select Configuration tab and make sure that the configuration file is not generated automatically. It should set to : '*Use an existing config.ini file as a template*' with the location: `${workspace_loc:launches/repoedit.win32/configuration/config.ini}`. '*launches*' is your imported launch folder from your SDK.
- Press run.



5.2.4 Launch PIM Database Setup

- Go to **Run > Run Configurations > Eclipse Application > Database Setup (win64)**.
- Select '*com.heiler.ppm.dbsetup.ui.product*' product.
- Make sure runtime JRE is set to a 64bit JRE or JDK.
- Select Plug-ins tab and make sure that '**Launch with**' is set to '*features selected below*'.
- Check '*com.heiler.ppm.feature.dbsetup.client.win64*' .
- Press run.
- For further steps to work with Database setup see: Server Database



5.3 Install examples

- Open **File > Import > Existing Projects into Workspace**.
- Select the C:\Informatica\PIM-SDK\examples directory for the root directory.
- Check all example projects, but the customizing.build and press **finish**

5.3.1 Examples at a glance

Project	Description
com.heiler.ppm.cu stom.rest	Examples for own rest based services. Contains a running example for a rest service which is able to return some item data based on EAN numbers
com.heiler.ppm.cu stomizing.activities	Examples how to disable UI contributions (like views, menu items etc.) using "Eclipse activities"
com.heiler.ppm.cu stomizing.article.u i.contributionclass	Example how to replace a standard view (like "Items #1") with a custom view using the extension point contributionClassProviders

Project	Description
com.heiler.ppm.customizing.article.ui.searchView	Example how to customize the standard "Search view" using the extension <code>point.searchViewAdditions</code>
com.heiler.ppm.customizing.ckeditor	Examples how to customize the standard RichText-Editor using extension <code>points.ckeditorConfiguration</code> , <code>richTextValidators</code> and <code>richTextMarkupConversionExtensions</code>
com.heiler.ppm.customizing.core	Core functionality of your own customizing, is empty by default. Use it as a template for your own customizing code
com.heiler.ppm.customizing.core_test	Test bundle for unit and integration tests. Contains an example for a unit test (<code>CustomizingUnitTest</code>) as well as for an integration test (<code>CustomizingIntegrationTest</code>). Additionally to that it contains working launch configurations for the given test suites. One for integration and one for unit tests. Use this bundle to implement your own tests for your customizing business code.
com.heiler.ppm.customizing.export.core	Examples for implementation of own export data providers, data types and export functions
com.heiler.ppm.customizing.export.core.test	Examples for writing tests for own export functionality
com.heiler.ppm.customizing.server	Server functionality of your own customizing, is empty by default. Use it as a template for your own customizing code
com.heiler.ppm.customizing.spelling.ui	Example how to use the <code>SpellCheckingAPI</code> to spellcheck the whole catalog in a mass-operation
com.heiler.ppm.customizing.ui	UI functionality of your own customizing, is empty by default. Use it as a template for your own customizing code
com.heiler.ppm.feature.customizing.*	Features for the client, server and tests (<code>JUnit</code>), used for the launch configurations and automatic builds. Add your own customizing plugins to these features
com.heiler.ppm.web.custom.theme	Sample project that demonstrates how to contribute a custom theme for WebAccess. Put your additional style definitions into the <code>styles.css</code> file.
com.heiler.ppm.web.custom.app	Sample project with some basic programmatic extensions of the PIM Web UI.
com.heiler.ppm.web.vaadin.sample	Sample web project that shows how to integrate a all new web interface by leveraging the PIM stack and the Vaadin framework.
customizing.build	Ready to use ant targets to build your customizing bundles and tests in an integrated and automated build system. Based on Eclipse p2 build infrastructure.