UMLMDA - Developerguide Guide

Dieter Moroff <moroff@user.sourceforge.net>

Table of Contents

1. Setting up Eclipse

$Revision: 1.4 $

This section describes how to setup eclipse 3.0 or higher for developing in the umlmda project.

To work on the RSM/RSA integration IBM Rational Software Modeller 6.0 or higher or IBM Rational Software Architect is required

1.1. Required / Recommended Plugins

Project-Set

Very useful to organize several projects.

Lomboz

The JSP-editor is used to edit the writer template file.

1.2. Configuring Eclipse

1.2.1. Classpath Variables

ANT_HOME

Points to the path where ant version 1.6 or higher is installed, required to build the umlmda ant tasks and types.

JBOSS4_HOME

Points to the path where JBoss version 4.0.2 or higher is installed.

HIBERNATE_HOME

Points to the path where hibernate version 2.1.6 or higher is installed, required to build the umlmda runtime components which support hibernate mapping. For the generator is is not required.

AXIS_HOME

Points to the path where AXIS version 1.2 or higher is installed. Required to test the generated web services.

1.2.2. File Associations

Jostraca Templates *.jtm

It is helpful the editing the Jostraca Generator Templates, which have a JSP like syntax, to use an editor which support jsp highlight especially to mark the difference between template code (outside <% %> tags) and writer code (inside <% %> tags.

Caution

With some versions of Eclipse and Lomboz starting a jsp editor is extremly slow, Eclipse 3.1M4 and Lomboz 3.1 Stream Build seams to be ok. Eclipse 2.1 and an older Version was ok too.

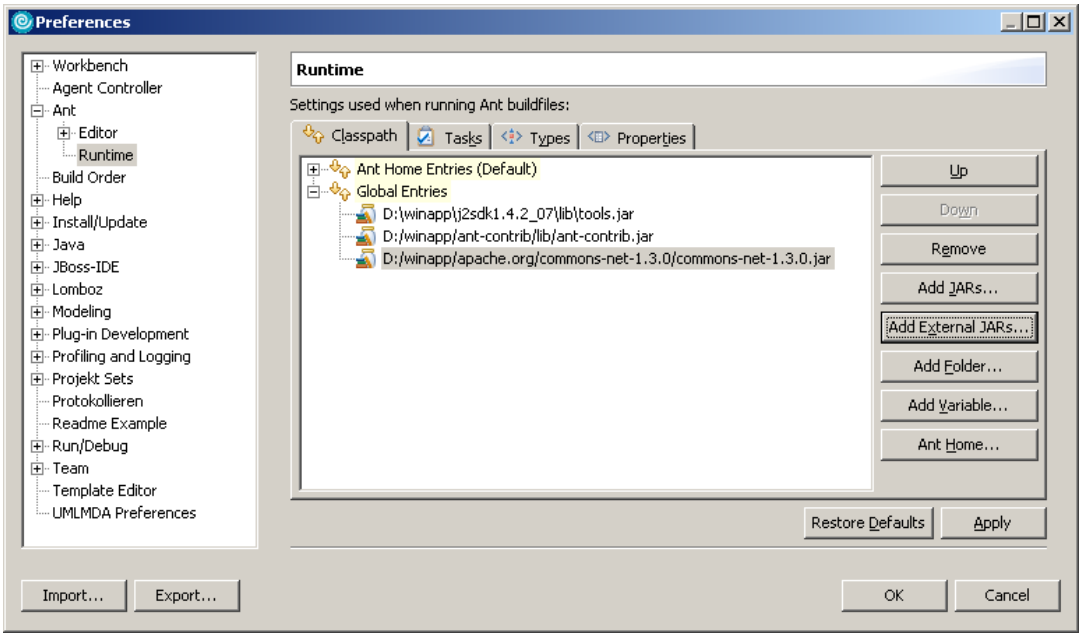
1.3. Configure Ant to build within Eclipse

To run the UMLMDA ant build files you have to add

- ant contrib task (from http://ant-contrib.sourceforge.net/ [???])
- Apache Commons Net (from http://jakarta.apache.org/commons/net/ [???])

to the ant classpath. Within eclise select Window|Preferences...|Ant|Runtime add the `ant-contrib.jar` and `commons-net-*.jar`.

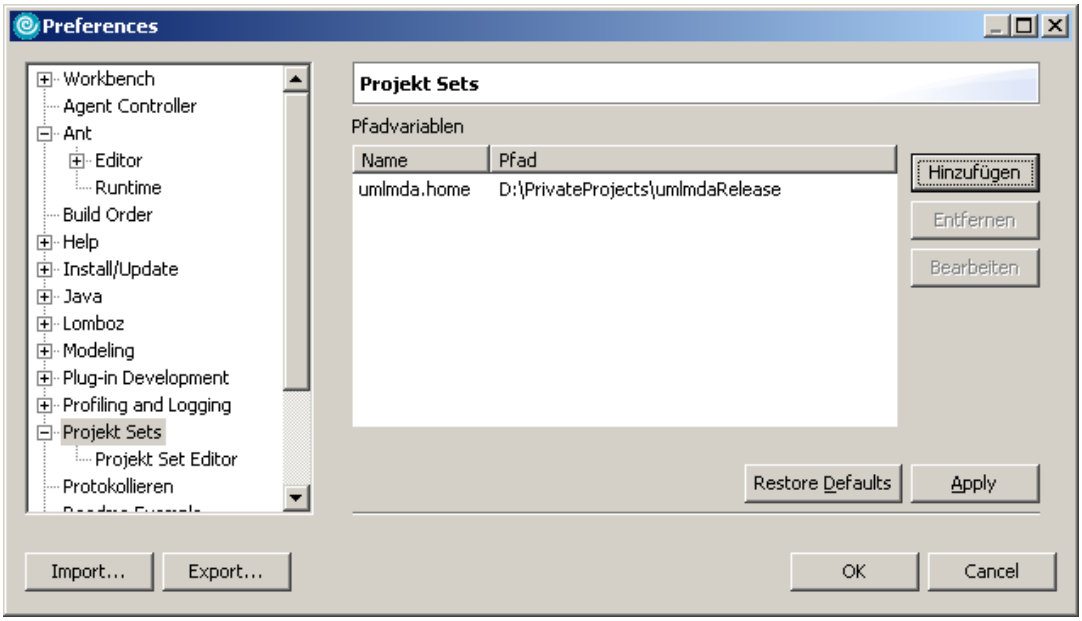
Figure 1. Eclipse Ant Runtime Preferences for UMLMDA builds.



1.4. Install UMLMDA within Eclipse

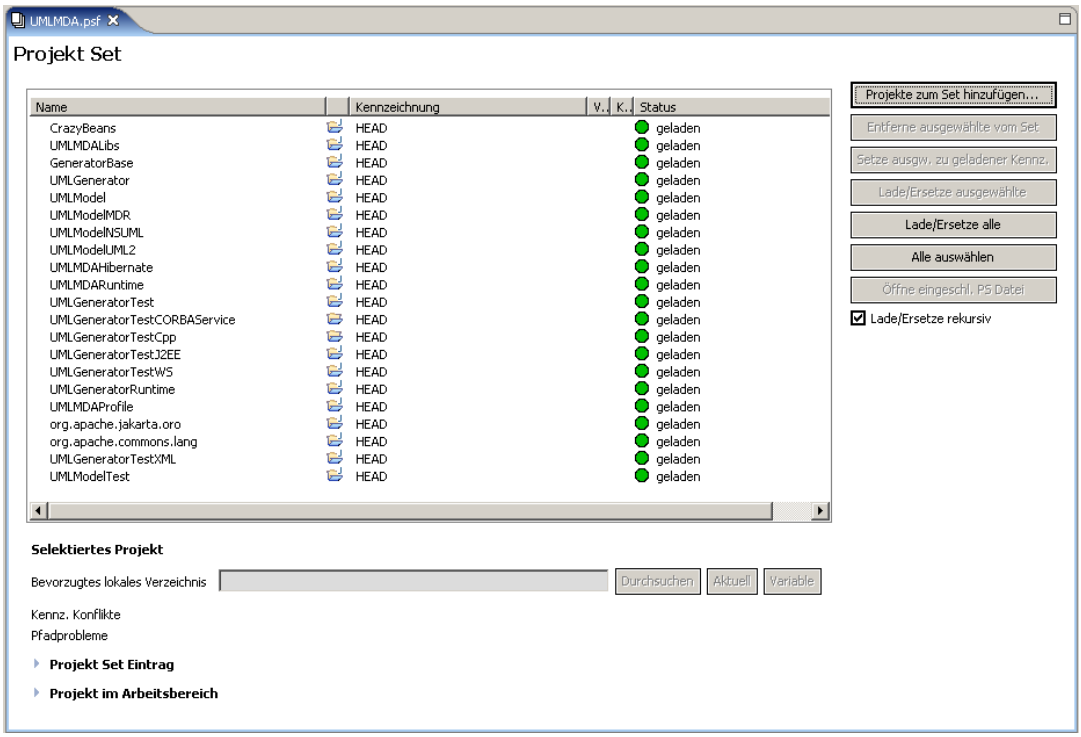
Install all projects from UMLMDA project set to the workspace where you want to build UMLMDA. To do what you have to set the preferences for the project set plugin and define the umlmda.home variable to the path where all projects are installed.

Figure 2. Eclipse Ant Projekt Sets Perferences for UMLMDA.



First you have to install the UMLMDA project from CVS. After that you can install the required project with the project set editor.

Figure 3. UMLMDA Project Set editor with loaded projects.

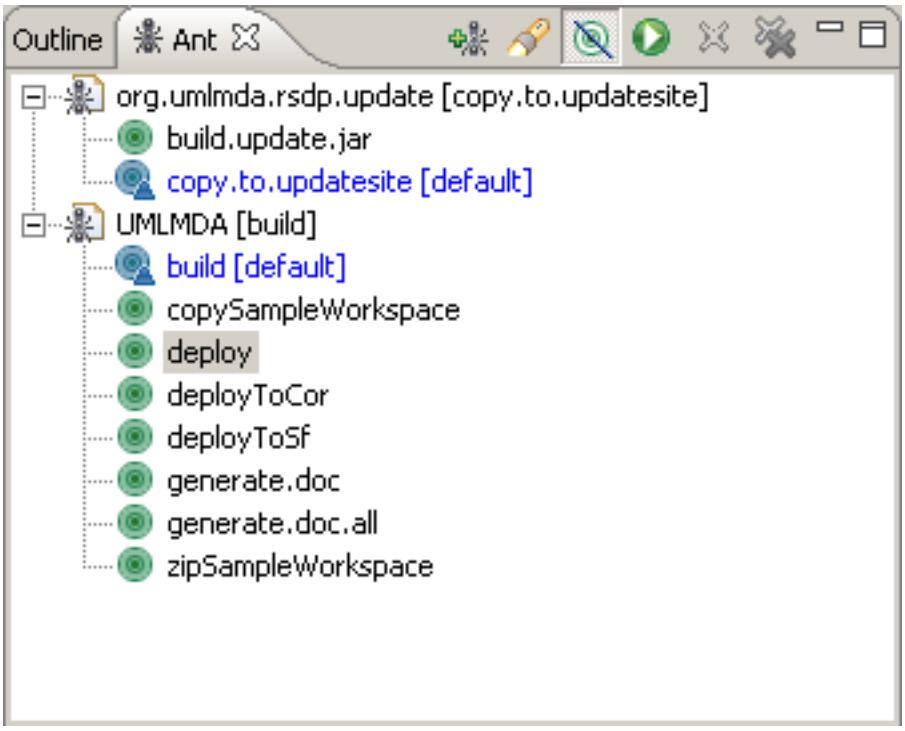


If you want to build the IBM Rational Software Modeler / Architect (RSM/RSA) integration. You have to install the project

1.5. Build UMLMDA within Eclipse

The best way to invoke the ant build from within eclipse is to open the ant view and add build.xml from the UMLMDA project and if RSM/RSA integration should be build the build.xml from org.umlmda.rsdp.

Figure 4. Ant view with the UMLMDA build files



1.5.1. Build the standalone deployment

To build the standalone deployment simply invoke the default target build in `build.xml` of the project UMLMDA. After sucessful build the deploy packages are created with the deploy target, these files can be uploaded to Sourceforge.net with the target deployToSf. The special target deployToCor is to run the build at COR AG[1] to deploy to the lokal deployment location.

1.5.2. Build the RSM/RSA integration deployment

Before buiding the RSM/RSA integration deployment verify that the new version is writen to the plugin.xml, feature.xml and Manifest.MF files of the plugins and features. Do to this you can use Search|File... and use the old version as search pattern.

In build.xml from the project org.umlmda.rsdp.update execute the following targets (build a standalone deployment before building the RSM/RSA integration to compile the writer templates):

- build.update.jar
- copy.update.size

[1]COR AG is my employer, remark Dieter Moroff

Add the new feature to the site.xml, remove old plugins and features from site.xml and the plugins and features directories. Deploy the updatesite to the internet.

1.6. Verify the build

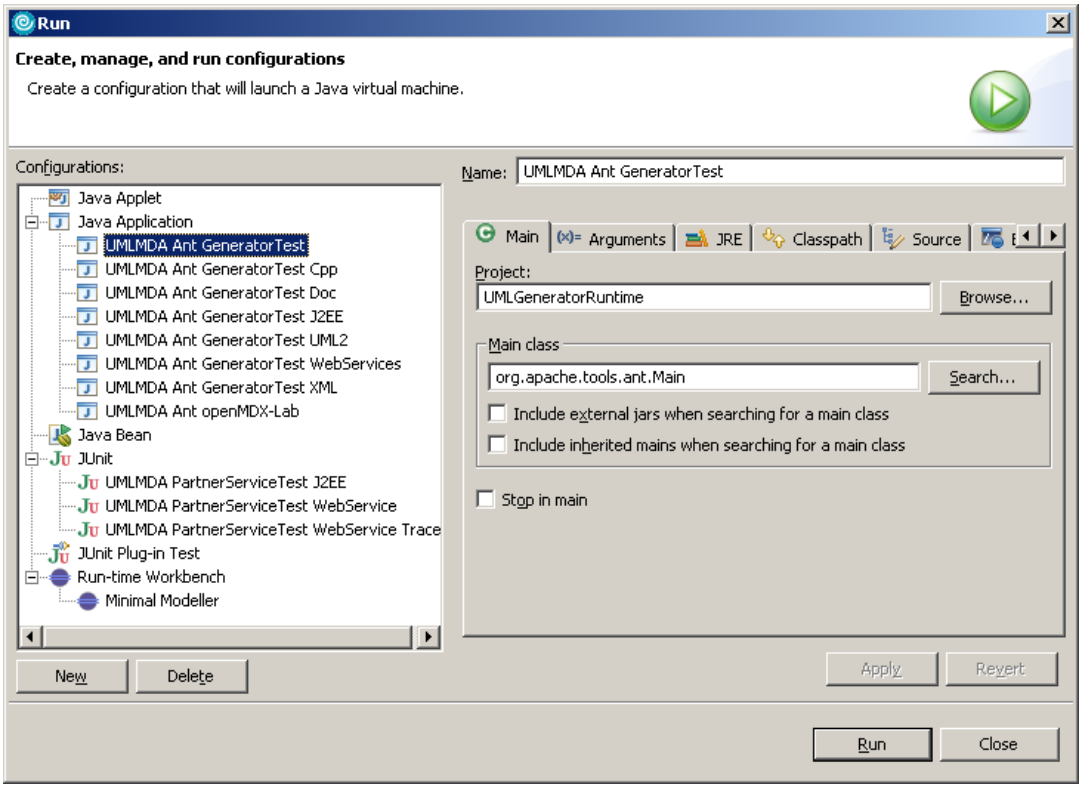
To verify the build there are some eclipse launch configuration to run test with the demo models. Before you run the test, ensure that all projects are refresh (to recompile the jostraca compiled template classes). After the tests are finished, make a refresh to the test projects UMLGeneratorTest* and verify that only desired changes occurred. Commit the desired changes to the CVS.

1.6.1. Lauch the tests for standalone deployment

To lauch the tests invkode from the eclipse launch menu the following lauch configurations, :

1. UMLMDA Ant GeneratorTest
2. UMLMDA Ant GeneratorTest J2EE
3. UMLMDA Ant GeneratorTest WebServices
4. UMLMDA Ant GeneratorTest XML
5. UMLMDA Ant GeneratorTest Cpp

Figure 5. Launch configuration with UMLMDA tests



1.6.2. Verify the generated sources

1.6.2.1. J2EE Test

1.6.2.2. Web service test

To test the generated webservice, the AXIS web service client stub have to be generated with the target generateWSStubs in `build.xml` of the project UMLGeneratorTestWS.

1.6.3. Check the changes from the tests

After refreshing the test projects (Select UMLGeneratorTest, UMLGeneratorTestCORBAService, UMLGeneratorTestCpp, UMLGeneratorTestJ2EE, UMLGeneratorTestWS and UMLGeneratorTestXML and press F5). To check the changes simple run a Team|Synchronize with repository ... on the selected test projects. If the changes are correct you can commit them to the CVS repositoy.

2. Extending UMLMDA

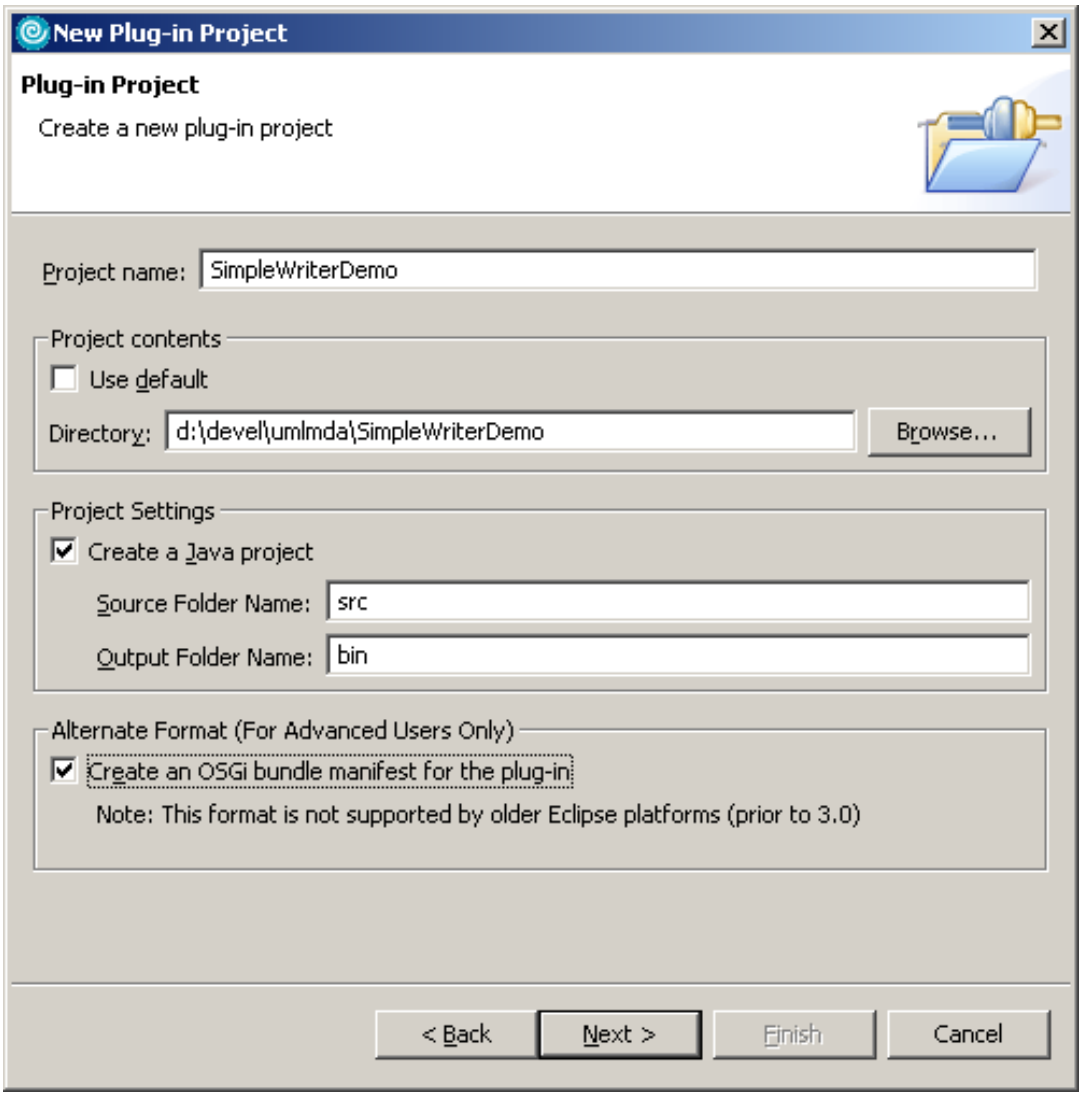
$Revision: 1.1 $

2.1. Writering own writers

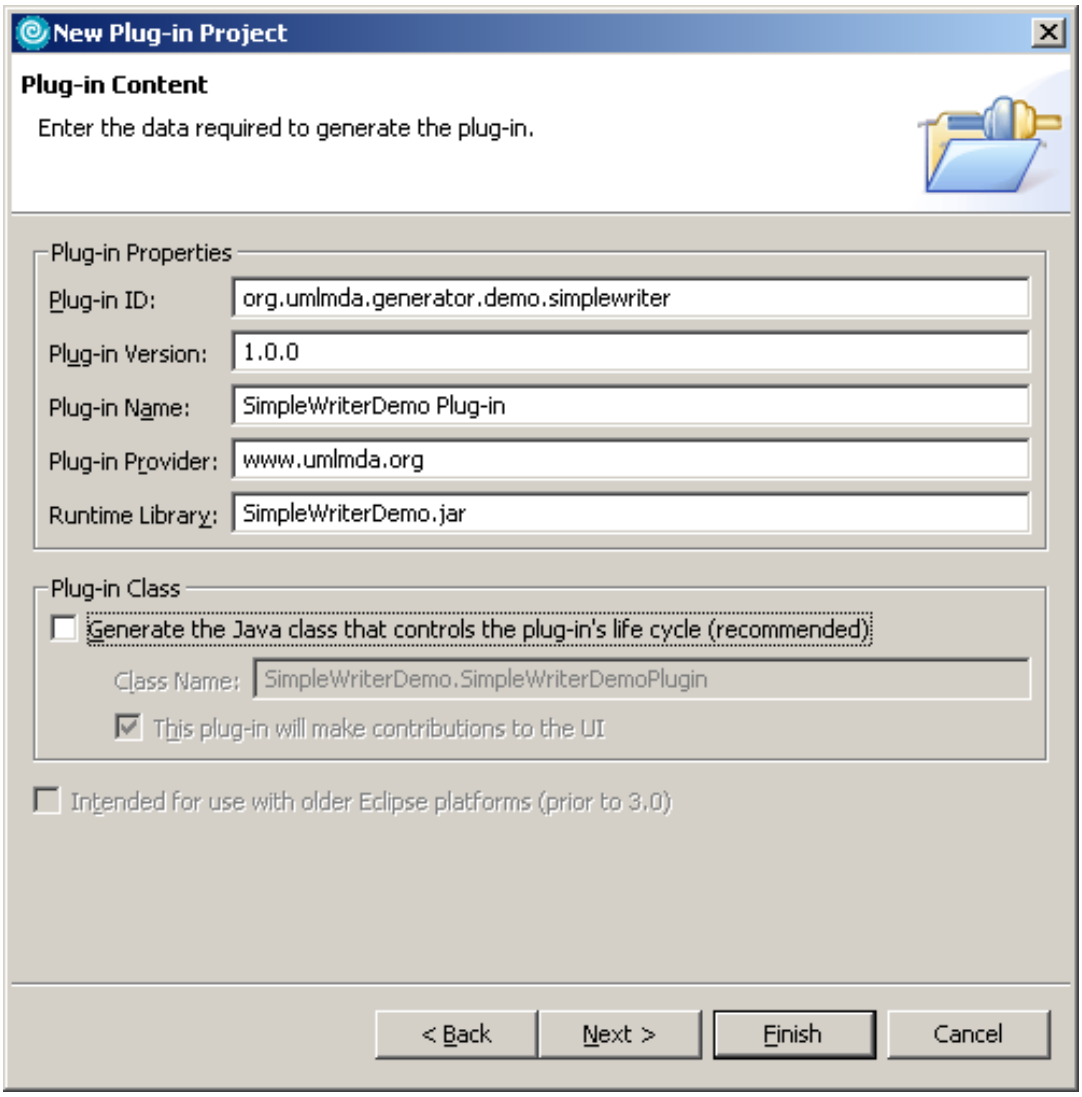
To write you own writers you create an Eclipse project in a workspace there at least the UMLMDA GeneratorBase project is installed (this project provides the Jostraca template compiler). Your own Jostraca writers can use writer baseclasses from the projects GeneratorBase or UMLGenerator.

2.1.1. Create a project for your writer

If you want to use the RSM/RSA integration (or a future UMLMDA eclipse integration) create a new Plugin project, otherwise create a new Java project.

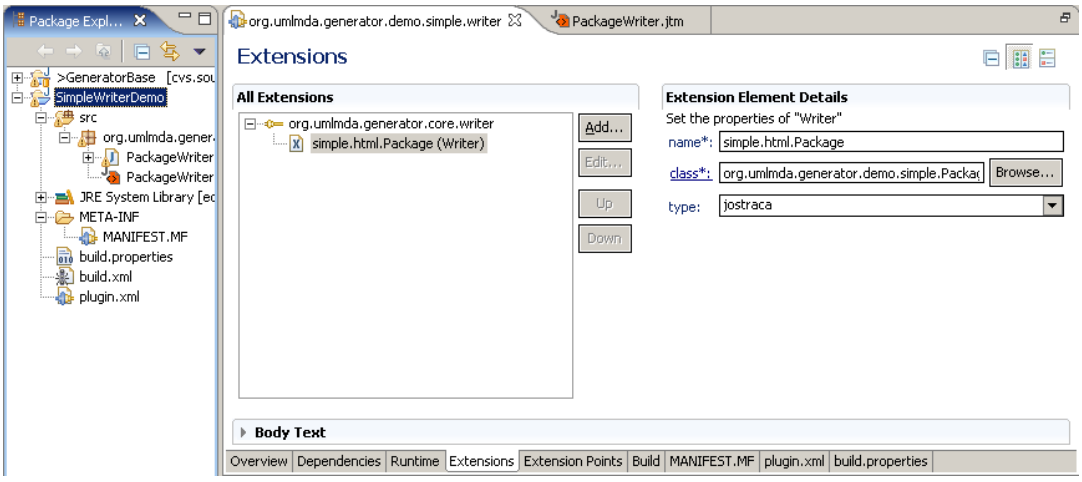


Check "Create an OSGI bundle manifest for the plugin".



Uncheck "Generate the Java class ...".

The plugin must depend on org.umlmda.generator.core, if base classes from UMLGenerator are desired org.umlmda.generator.uml must also be referenced.



In the plugin.xml editor for every writer an extension element on the extension point `org.umlmda.generator.core.writer` must be added.

2.2. Deploy writers in an own plugin for usage with RSM/RSA integration

Because the RSM/RSA integration using the Eclipse plugin technology any extension must be a plugin an implements the extension point `org.umlmda.generator.core.writer` published by UMLMDA. A sample of implementing a custom writer is the openMDX-Generator[2] . In the plugin.xml the implemented writers are declared in the extension point, to make then accessible to the UMLMDA generator.

```
<?eclipse version="3.0"?>
<plugin>
  <extension point="org.umlmda.generator.core.writer">
    <Writer type="jostraca"
            class="ch.css.umlmda.generator.writer.openmdx.sql.SlicedDDLWriter"
            name="openMDX.sql.DDL"/>
    <Writer type="jostraca"
            class="ch.css.umlmda.generator.writer.openmdx.xml.DeploymentWriter"
            name="openMDX.xml.Deployment"/>
  </extension>
</plugin>
```

Currently there is only an extension point for writers and only jostraca type writers are supported. In future versions JET based templates and extending own adapter factories will be supported.

2.2.1. Packaging own writers

The own writers should be packaged and deployed in a Eclipse feature and an update site.

3. Tips and Tricks

$Revision: 1.1 $

3.1. Using the Updatemanager with the same version

While developing a plugin it's sometimes useful to install the plugin with the update manager, rather than testing it with the runtime workbench. If the feature or plugin version number doen't change it's cached in the configuration directory of eclipse. To handle this you can simple change the plugin version number every time you install (which might be not very reasonable) or you simply delete all subdirectories in the `configuration` directory of eclipse installation.

[2]There are writers to generate deployment descriptors and DDLs from openMDX [http://www.openmdx.org] models. The writers are develeped for the CSS Health Insurance [http://www.css.ch], Switzerland and currently not open source.