

## **Installation of openArchitectureWare 4.2 for Model with Enterprise Architect**



**2008-01-07**

Copyright © 2007 – 2008 Ulrich Brawand

Copies of this document may be made for your own use and for distribution to others, provided that you do not charge any fee for such copies and further provided that each copy contains this Copyright Notice, whether distributed in print or electronically.

## Table of Contents

Overview.....	4
Installation of Eclipse 3.3.....	4
Installation of UML 2.1 and Eclipse Plug-in Development Component.....	4
Installation of oAW 4.2.....	7
Configuration of oAW .....	9
Installation of UML2Exporter for EA .....	9
Create User Library for UML2Exporter .....	9
Create and Configure Project .....	10
Create Project structure .....	11
Example Workflows for Working with EA and oAW .....	11
Further Information, Bugs and Feature Request .....	13

### Overview

This article describes how to install all components to work with openArchitectureWare 4.2 and Enterprise Architect from SpraxSystems as the UML modeling tool.

Following components are required:

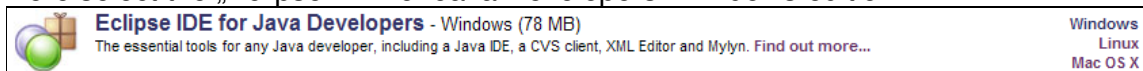
- Eclipse 3.3.x Europa Edition
- Eclipse Java Plug-in Development Component 3.3.2
- Eclipse WST 2.0.1
- Eclipse Modeling Framework EMF 2.3
- Eclipse UML 2.1
- oAW 4.2
- UML2Exporter for EA

So let's start.

### Installation of Eclipse 3.3

First we have to download the correct Eclipse distribution for working with oAW 4.2. Open site <http://www.eclipse.org/downloads>

Here select the „Eclipse IDE for Java Developers” Windows edition.

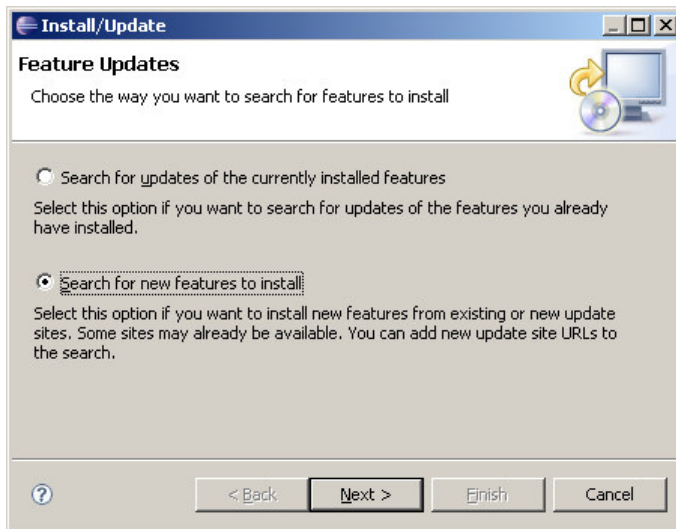


Unzip the file in a directory of your choice. We call it now <ECLIPSE\_PATH>.

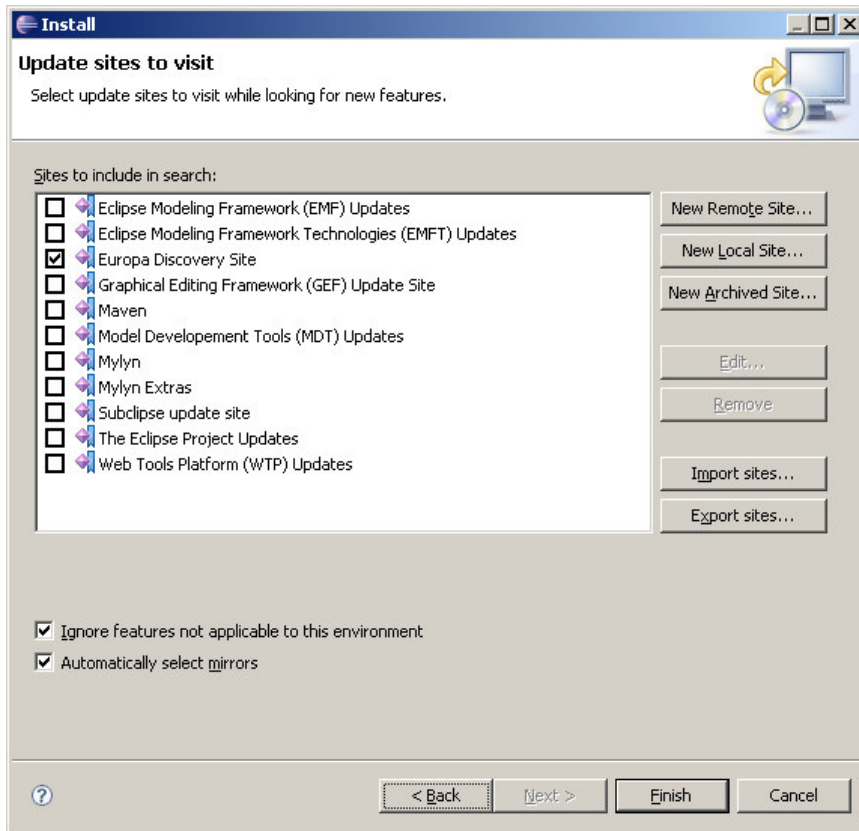
With this distribution EMF 2.3 and WST 2.0.1 are already installed.

### Installation of UML 2.1 and Eclipse Plug-in Development Component

Open “Help – Software Updates – Find and Install ...”



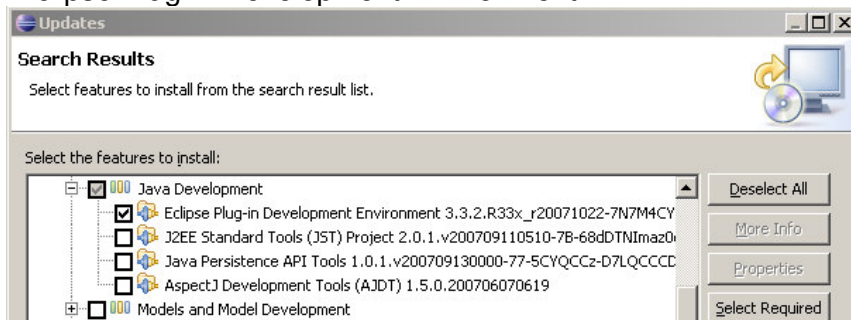
... and select „Search for new features to install“ and press „Next“.



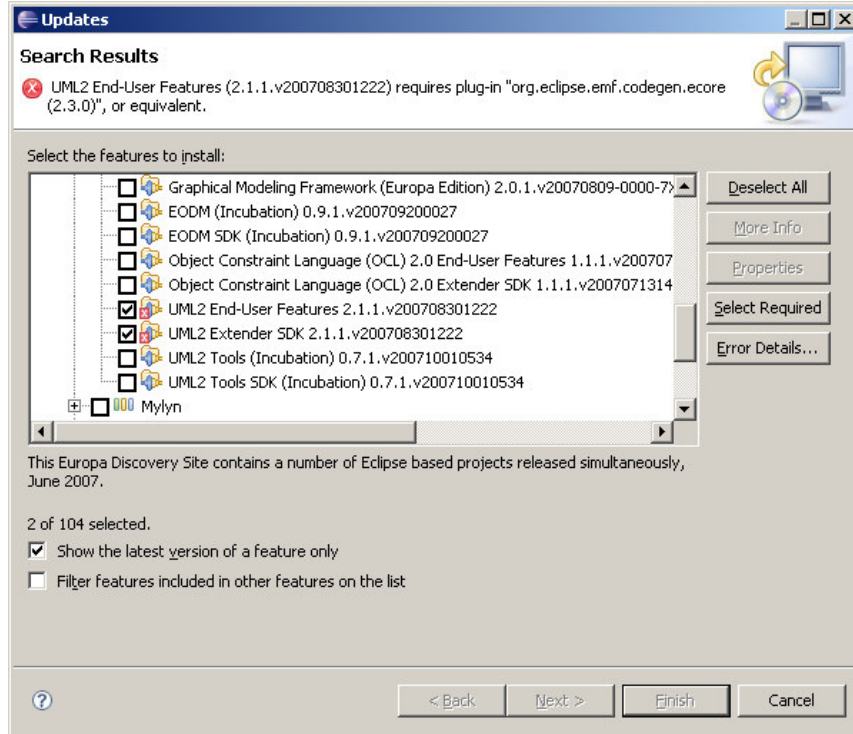
Here select „Europa Discovery Site“ and press „Finish“.  
Select the download site of your choice, if select screen appears.

**Hint:** If connection could not be established, check if a proxy definition is required.

Now expand the node „Europa Discovery Site“, expand “Java Development” and select “Eclipse Plug-in Development Environment ..”.

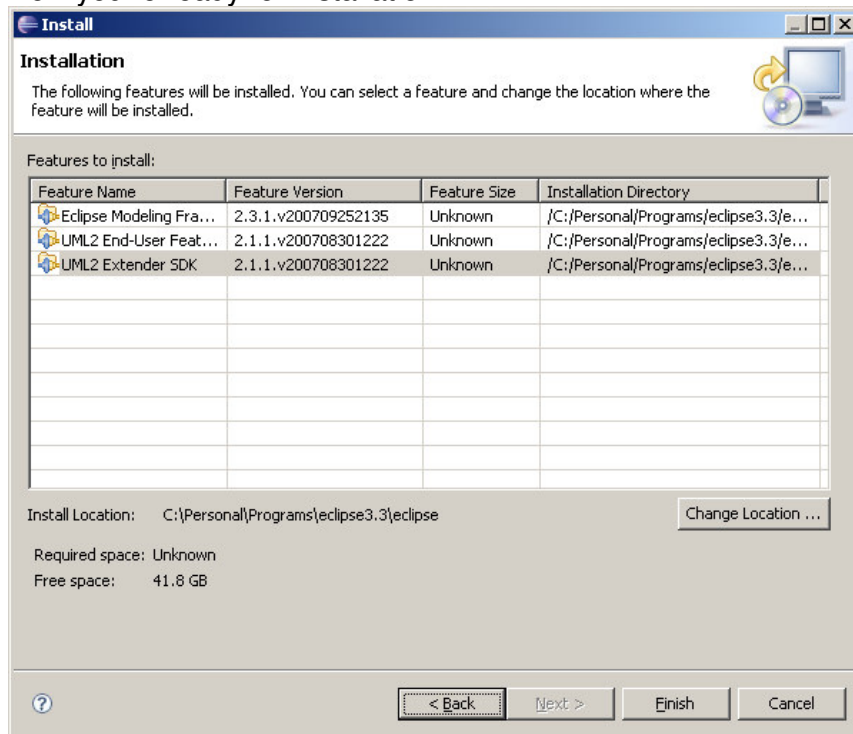


Expand „Models and Model Development” and select „UML2 End-User Features 2.1.1” and „UML2 Extender SDK 2.1.1”.



Now an additional plugin “org.eclipse.emf.codegen.ecore” is required. Press button “Select Required” and then “Next”. Accept the terms in license agreements and press “Next” again.

Now you're ready for installation.





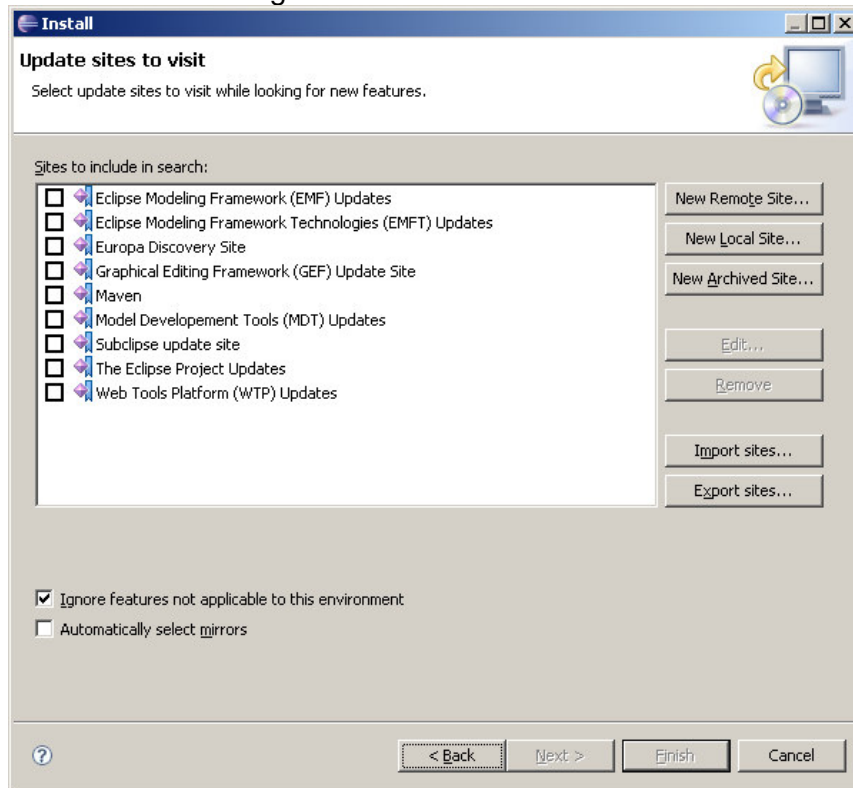
Press “Finish” and commit installation of features and restart of Eclipse.

Now all required components of Eclipse are installed and we are ready for openArchitectureWare.

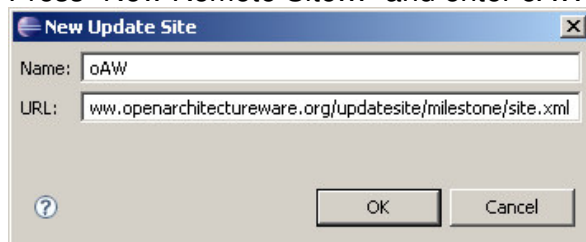
### Installation of oAW 4.2

The easiest way of installing openArchitectureWare is by the update site.

Open “Help – Software Updates – Find and Install ...” and select “Search for new features to install” again.



Press “New Remote Site...” and enter oAW as name

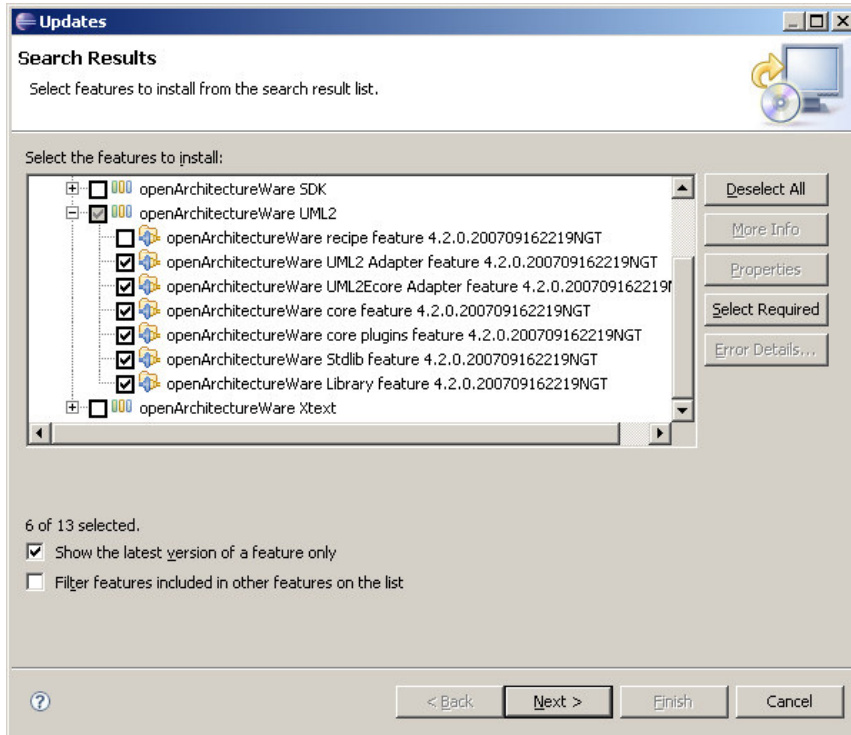


and <http://www.openarchitectureware.org/updatesite/milestone/site.xml> as URL and press „OK“.

The new entered site is select. Press „Finish“.

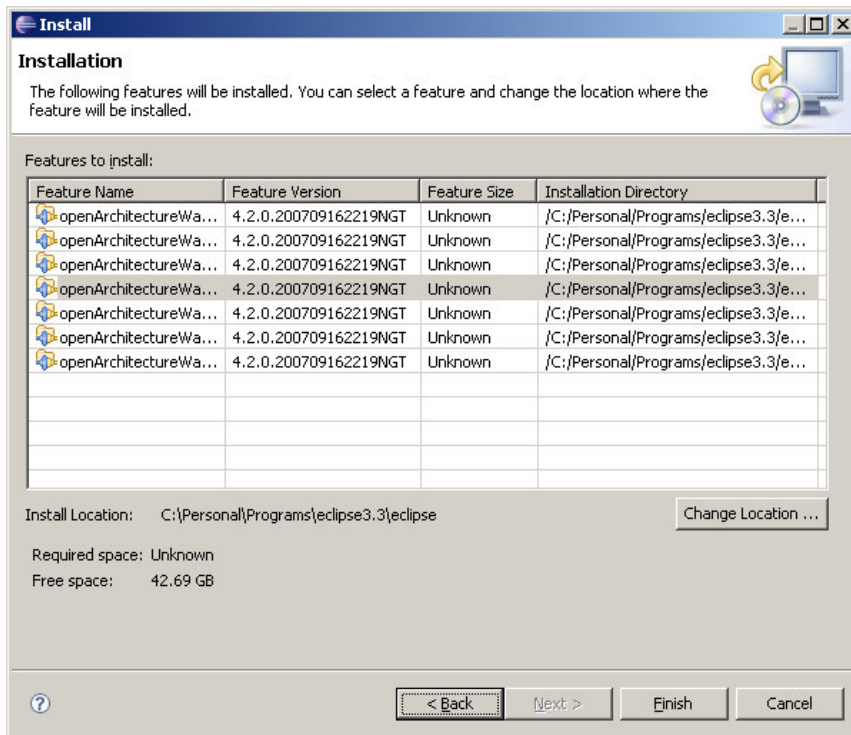
Select “UML2” and unselect “openArchitectureWare recipe feature” if you don’t need it.

## Installation oAW4.2 with EA



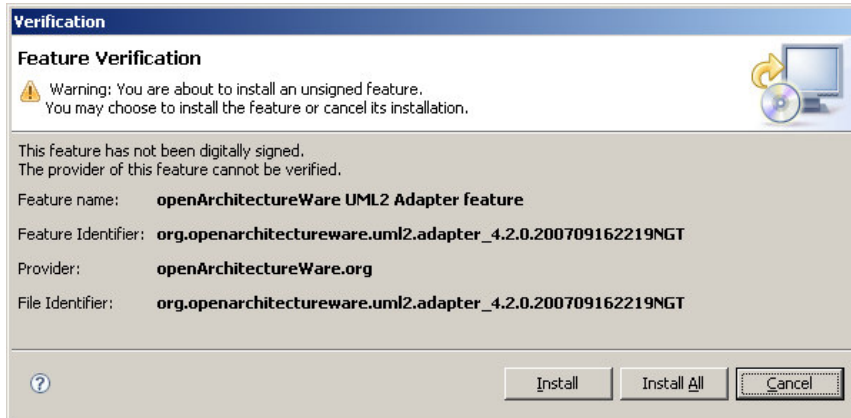
If you plan to work with Xtext, select this option as well and press „Next“.

Accept license agreements and press „Next“ again.



Now you're ready for installation and you can press „Finish“.

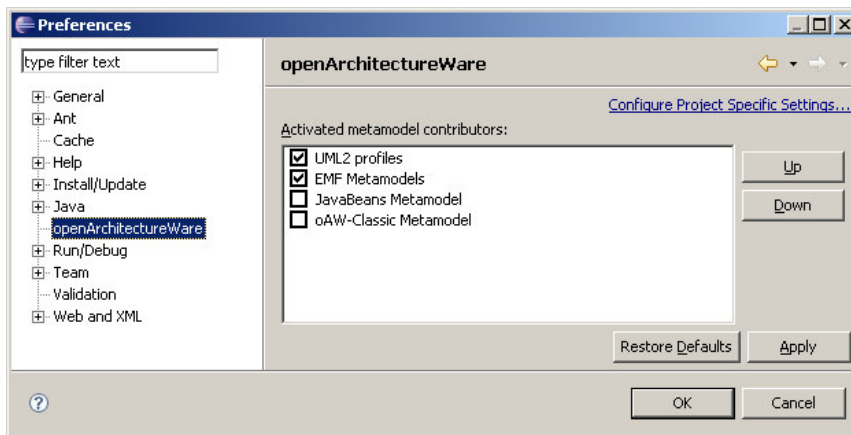




Commit this dialog with „Install All“ and restart Eclipse.

## Configuration of oAW

Now open „Windows/Preferences“ and select „openArchitectureWare“.



Activate „UML2 profiles“ and make UML2 the first contributor in the list.

## Installation of UML2Exporter for EA

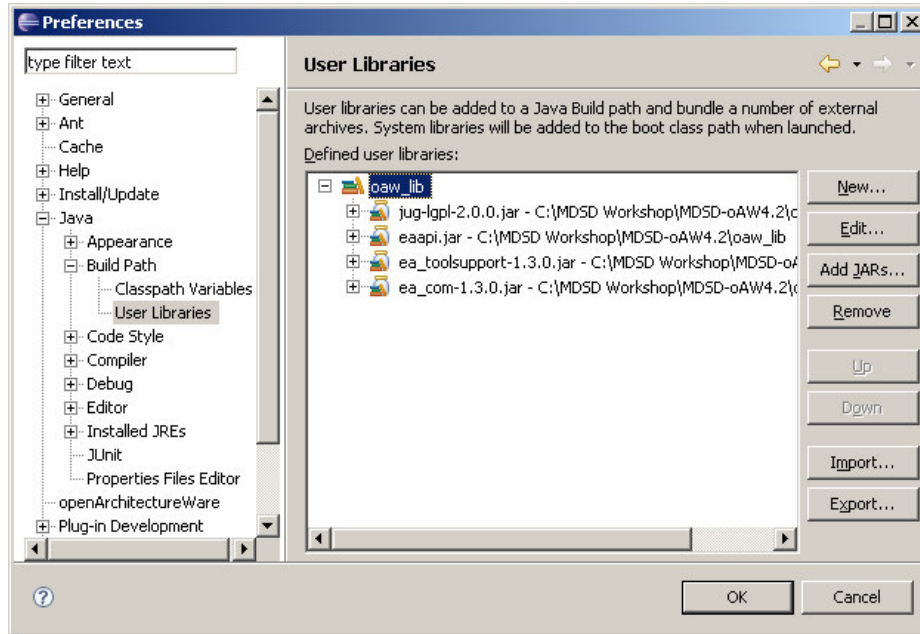
For exporting models from EA in EMF based XMI files the UML2Exporter for EA is required. Since release 1.3.0 the export interface uses the Java API of EA.

Download actual version of the exporter from download section of components4oaw ([http://sourceforge.net/project/showfiles.php?group\\_id=203960](http://sourceforge.net/project/showfiles.php?group_id=203960)) and unzip all files in an own directory.

### **Create User Library for UML2Exporter**

For simple usage of UML2Exporter in projects define a User Library in Eclipse. Select „Windows/Preferences ...“ and here select „Java / Build Path / User Libraries“.

Create a new library with „New...“ and name it „oaw\_lib“ (or whatever you like).



From the previously extracted files add the libraries

- jug-igpl-2.0.0.jar
- eaapi.jar
- ea\_toolsupport-<version>.jar
- ea\_com-<version>.jar

You will find the java archive eaapi.jar in the subfolder „Java API“ of your EA installation.

---

**Important:** Add the file *SSJavaCOM.dll* to the System32 directory or add it to the root of every oAW project in Eclipse.

---

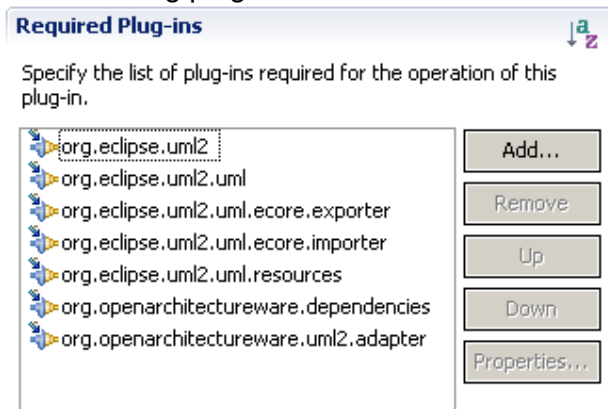
## Create and Configure Project

Now we are ready to create a project in the current workspace and configure it to work correctly with UML2 models.

Following steps are required:

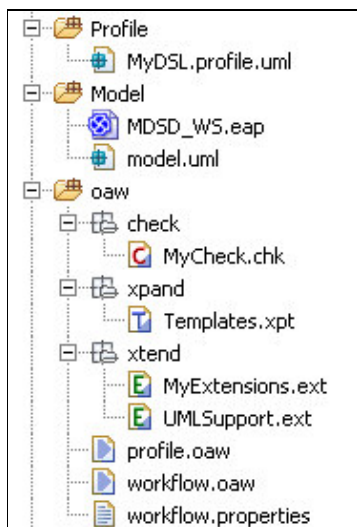
- Create a new Java Project and name it as you like
- Convert it to a plug-in project (*PDE Tools* → *Convert Project to Plug-in Projects*)
- Open file *MANIFEST.MF* in *META-INF* directory and switch to *Dependencies*

- Add following plug-ins:



### Create Project structure

I use the following structure in my projects



Profile: Contains the profile file created with profile workflow.

Model: \*.eap and \*.uml files are stored here

oaw: Contains all the files used for checking, transforming and code generation spread in the subfolders *check*, *xpanse* and *xtend*.

The file UMLSupport.ext is part of the UML2Exporter. You will find it in the ZIP-file.

### Example Workflows for Working with EA and oAW

Here are the example workflows you can use for your project (workflow.oaw). Both workflows could be configured in the workflow.properties file (see below).

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow>

  <property file="workflow.properties" />

  <bean class="oaw.uml2.Setup" standardUML2Setup="true" />

  <!-- Metamodel-Definition -->
  <bean id="EmFMM" class="oaw.type.emf.EmfMetaModel">
    <metaModelPackage value="org.eclipse.emf.ecore.EcorePackage"/>
  </bean>
</workflow>
```

```
</bean>
<bean id="UmlMM" class="oaw.uml2.UML2MetaModel"/>
<bean id="${profile_name}MM" class="oaw.uml2.profile.ProfileMetaModel">
  <profile value="${profile_dir}/${profile_name}.profile.uml" />
</bean>

<component class="oaw.uml2.toolsupport.ea.EA_Xmi2Exporter" >
  <EapFile value="${project_root}${ea_file}" />
  <ModelFile value="${model_file}" />
  <PackageName value="${model_pkg}" />

  <profile value="${project_root}${profile_dir}/${profile_name}.profile.uml" />
  <PackageProfile name="${apply_pkg}" value="${profile_name}" />

  <changedOnly value="true"/>
  <Cleanup value="true"/>
</component>

<component class="oaw.emf.XmiReader">
  <modelFile value="${model_file}" />
  <outputSlot value="${output_slot}"/>
</component>

<if cond="${doCheck}">
  <component class="oaw.check.CheckComponent">
    <metaModel idRef="EmfMM"/>
    <metaModel idRef="UmlMM"/>
    <metaModel idRef="${profile_name}MM"/>

    <checkFile value="${check_file}" />
    <expression value="${output_slot}.eAllContents" />
  </component>
</if>

<component id="dirCleaner"
  class="oaw.workflow.common.DirectoryCleaner"
  directories="${gen_dir}"/>

<component id="generator" class="oaw.xpand2.Generator" skipOnErrors="true">
  <fileEncoding value="ISO-8859-1" />

  <metaModel idRef="EmfMM"/>
  <metaModel idRef="UmlMM"/>
  <metaModel idRef="${profile_name}MM"/>

  <expand value="${expand}"/>

  <outlet path="${gen_dir}"/>
  <postprocessor class="oaw.xpand2.output.JavaBeautifier"/>
  <postprocessor class="oaw.xpand2.output.XmlBeautifier"/>
</outlet>
</component>
</workflow>
```

### workflow.oaw

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow>
  <property file="workflow.properties" />

  <bean class="oaw.uml2.Setup" standardUML2Setup="true" />

  <component class="oaw.uml2.toolsupport.ea.EA_Xmi2ProfileExporter" >
    <EapFile value="${project_root}${ea_file}" />
    <ModelFile value="${project_root}Profile/${profile_name}.profile.uml" />
    <PackageName value="${profile_pkg}" />
    <Cleanup value="true"/>
  </component>
</workflow>
```

### profile.oaw

```
# Property file
project_root =
ea_file      = Model/MSDD_WS.EAP
```

```
model_file      = Model/model.uml
model_pkg       = Models/Project Model/ExamplePkg
output_slot    = model

# Profile definition
profile_pkg     = Models/Profile Model/mydsl
profile_name    = mydsl
apply_pkg      = ExamplePkg
profile_dir     = Profile

# Checkfile
doCheck        = false
check_file     = check::MyCheck

# Generator
expand         = xpanse::Templates::Root FOR model
gen_dir        = src-gen
workflow.properties
```

## Further Information, Bugs and Feature Request

You will find more information here:

[www.openarchitectureware.org](http://www.openarchitectureware.org)

[www.sparxsystems.com](http://www.sparxsystems.com)

<http://uml2ea.blogspot.com>

<http://components4oaw.sourceforge.net>

If you find a bug or have a feature request, pls submit it here:

[http://sourceforge.net/tracker/?group\\_id=203960](http://sourceforge.net/tracker/?group_id=203960)