

oAW 4 Installation

Markus Voelter, voelter@acm.org, www.voelter.de



Table of Contents

| | |
|----------------------------------|----------|
| DOWNLOAD..... | 3 |
| PREREQUISITES..... | 3 |
| MINIMAL INSTALLATION..... | 3 |
| PLUGINS..... | 3 |
| CORE..... | 4 |
| CLASSIC INSTALLATION..... | 4 |
| RECIPE INSTALLATION..... | 5 |
| ADAPTER INSTALLATION..... | 6 |
| DOCUMENTATION..... | 7 |

Download

From the openarchitectureware.org site you can find links to download the various oAW4 packages. Please get the respective ZIP files from there. The following table explains the contents of the various packages; for each package, we distribute a single ZIP file.

| Package | Description | Depends on ... |
|-------------------|--|----------------|
| oaw-core | Workflow Engine, EMF integration, Xpand 2 Engine, Wombat Engine | oaw-core |
| oaw-core-plugins | Eclipse-based IDE support, including the Xpand 2 Editor | |
| oaw-classic | Support for the classic oAW, incl. Metamodel Generator, the "old" instantiators, IDE support | oaw-core |
| oaw-adapter | Adapters to various 3 rd party tools such as OCL checks and ATL integration | oaw-core |
| oaw-recipe | Recipe Framework incl. IDE support | |
| oaw-documentation | Docs | |

Prerequisites

This tutorial expects that you use oAW4 with Eclipse. This is not strictly necessary, however, as a consequence of the Eclipse IDE integration it certainly makes a lot of sense. Consequently, we expect that you have **Eclipse 3.1** installed, including **EMF 2.1**. You will have to install additional packages (such as Eclipse GMT ATL or Kent OCL) for specific scenarios. Take a look at the *oclExample* and *atlExample* documentations for details on what to install, and how.

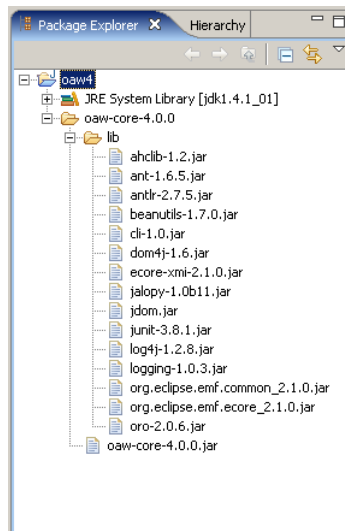
Minimal Installation

Plugins

As a first step, please download the *oaw-core-plugins* package. Extract the contents into your Eclipse directory. The ZIP contains a *plugins* and a *features* directory that contains all the necessary oAW core plugins. Extracting the ZIP will put them into the correct place in you Eclipse installation.

Core

In addition to the plugins, you need a couple of JAR files accessible to your application projects. We propose to install them into a project in your Eclipse workspace. You might want to call that project *oaw4*. Then please extract the *oaw-core* package into that project. This will result in the following workspace structure.



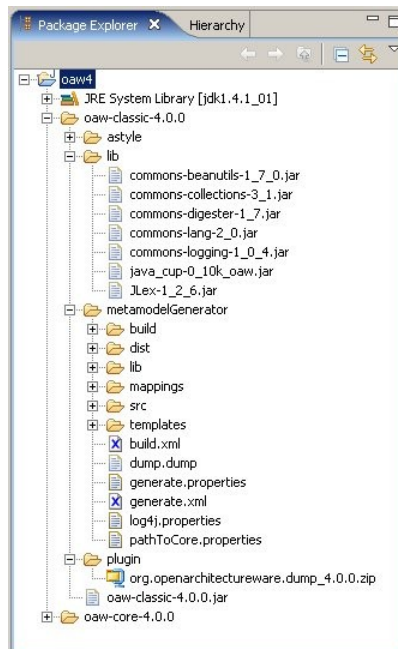
You are now ready to run the first examples.

Classic Installation

Note that the documentation for *oaw-classic* assumes that you have been working with version 3 of openArchitectureWare before and are familiar with oAW 3 concepts and tools.

Beside the “old” stuff, the *classic* package contains also the whole UML/XMI and XML instantiator handling.

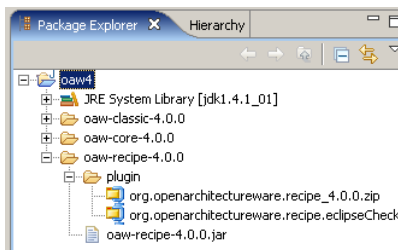
In case you want to use oAW 4 Classic, you have to download the *oaw-classic* package. Again, make sure you extract that ZIP file into the same workspace project you created above. As a consequence, your workspace (and the *oaw4* project) should look as follows:



The *oaw-classic-4.0.0/astyle* directory contains the astyle pretty printer that you might want to use after code generation. Inside the *lib* directory, you'll find the libs which the *oaw-classic* requires in addition to those already delivered by *oaw-core*. In *metamodelGenerator*, you can find the metamodel generator that should be well known from version 3 – it's basically unchanged. Finally, the *plugin* directory contains the dump browser plugin. Please open the ZIP file and extract it into your Eclipse directory.

Recipe Installation

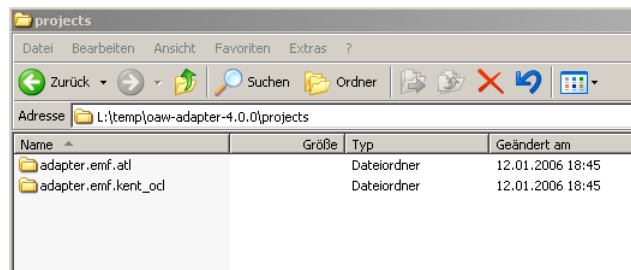
The recipe framework is independent of oAW 3 or 4. To install it, please download the *oaw-recipe* package. Again, please extract the package into the *oaw4* project in your workspace. That project looks as follows:



To install the IDE support, please extract the two ZIP files in the *plugin* directory into your Eclipse installation. Again, the ZIP files contain the *plugins* directory, so make sure you extract the ZIPs into the Eclipse root directory.

Adapter Installation

The adapter package is called *oaw-adapter*. Please download it from the openarchitectureware.org web site. The respective ZIP file should be extracted **not** into the *oaw4* project in the workspace. Rather, extract it to some temporary directory. This then looks as follows:



Depending on the adapters you actually need, the two (or in future releases, more) projects, should be copied into your application development Eclipse workspace, right **next** to the *oaw4* project we used before, not into it! Also, you might need additional Eclipse plugins (such as ATL) for the adapters to work. Please take a look at the respective tutorials for details (*atlExample*, *oclExample*).

After importing these two new projects, you might get a number of compile errors, because the dependencies in these projects cannot be resolved; the respective Eclipse classpath variables have to be set. The following table shows, which variables you have to set, and where they have to point to:

| Variable | ... points to |
|----------|-------------------------|
| OAW_CORE | oaw4/oaw-core-4.x.x |
| OAW_LIB | oaw4/oaw-core-4.x.x/lib |

Also, to use the ATL adapter, you have to install ATL first. Take a look at the *atlExample* documentation for details. For using the Kent OCL adapter, you have to install Kent OCL. See the *oclExample* for details.

Documentation

To have easy access to the documentation, you might want to install the docs into your *oaw4* project, too. Download the respective package and extract it into the *oaw4* project in your workspace. Since you are already reading this document, it's likely that you have already downloaded the documentation package ☺

About our Sponsors

itemis GmbH & Co. KG is an independent IT service company with an emphasis on consulting, coaching, and software development. Every single itemis expert provides many years of project experience and widespread knowledge about all object oriented and component based software development issues - especially in the field of model driven software development.

b+m is the founder of the openArchitectureWare project. The software was originally developed within the scope of many successful projects. b+m opened the software to the community in late 2003. All of the paradigms of Model-Driven Software Development including Product Line Engineering and not only the generator framework have become a key concept for product and customer specific development at b+m. b+m customers can make use of long time experience and substantial know-how in that field. Located at the company headquarters in Melsdorf/Kiel and at its subsidiaries in Berlin, Cottbus, Hamburg, Hanover and Kiel the b+m staff of 205 provides practical solutions for customized business applications, business process optimization and comprehensive architecture, project and quality management.

oose Innovative Informatik GmbH offers coaching, consulting and training in all themes about software engineering. The main focus of their activities are software architecture, requirements engineering and project-management. oose have first-hand information and experience, because our staff take actively part with others in actual trends, standards and innovations. Our staff support this and pass their know-how regularly on by writing and publishing books or being speaker at conferences, etc. Within the OMG oose collaborate actively on the specifications of the UML and also the SysML.

MID Enterprise Software Solutions GmbH is a leading supplier of optimized tool environments for standardsbased and model-centric software development as well as business process modeling. This includes professional tool consulting and tool components to build a complete tool environment using the best techniques and tool modules available - Architectural and Operational Excellence. With innovatorAOX, MID provides a holistic standard tool environment for object- and function-oriented software development as well as business process and data modeling to help its customers establish highly efficient processes and tool environments for software production, The unique and seamless integration of business process modeling into the development process ensures an unprecedented level of convergence of business requirements and implemented IT systems. Project members from all departments speak the same language and all requirements are clearly described.