



SIDE
MDA Designer Studio

Installation Guide
(Version 1.3)

Copyright, Trademark, and Version Notices

Copyright Notice Copyright © 2005-2011 BlueXML

All rights reserved. Information relating to the architecture and software described in this guide is proprietary to BlueXML and, as applicable, its licensors. Rights to use the software and the accompanying documentation are governed by a separate license agreement from BlueXML presented prior to software installation. Information in this document is subject to change without notice. No part of this guide may be reproduced or published in any form or by any means, electronic or mechanical, including photocopying, for any purpose other than the purchaser's personal use.

Trademark Notice

BlueXML, SIDE, SIDE-Labs are trademarks or service marks of BlueXML. Other company and product names may be the trademarks or service marks of third parties. Third party software is installed as distinct, unmodified software programs during the SIDE platform installation process. Licensing information pertaining to those third party programs is available on the SIDE Installation package.

Document and Software Versions

- Document Version 1.x
- Software Version 1.x & 2.x

TABLE OF CONTACT

INTRODUCTION.....	4
WHAT IS SIDE ?.....	4
TECHNOLOGICAL OVERVIEW.....	6
SIDE DOCUMENTATION STRUCTURE.....	7
TYPOGRAPHIC CONVENTIONS.....	7
PREDEFINED PARAMETERS.....	7
ABOUT THIS MANUAL.....	9
SIDE PRODUCT ARCHITECTURE.....	10
SIDE INSTALLATION PROCEDURE.....	13
JAVA INSTALLATION.....	13
<i>Step 1: Download and install JDK 1.5/1.6.....</i>	<i>13</i>
For Windows Platform.....	13
For Linux Platforms.....	13
<i>Step 2: Configure JDK 1.5/1.6.....</i>	<i>13</i>
On Windows Platform.....	13
On *n*x Platform.....	14
ECLIPSE INSTALLATION.....	14
<i>Step 1: Download SIDE</i>	<i>14</i>
<i>Step 2: Configure SIDE</i>	<i>14</i>
<i>Step 3: Start SIDE</i>	<i>16</i>
CHECKLIST.....	17
SIDE INSTALLATION TEST.....	18
SIDE INSTALLATION EXTENSIONS.....	24
INSTALLATION OF PLUG-INS.....	24
CONFIGURATION OF MAVEN PROJECTS.....	27
SIDE FEEDBACK.....	30
TARGETED FRAMEWORKS INSTALLATION.....	31
ALFRESCO.....	31
BIRT.....	31
CHIBA.....	31
FACETMAP.....	31
EXTJS VIEW AND FORMS DEMO.....	31
WORKFLOW DEMO.....	32



INTRODUCTION

What is SIDE ?

SIDE is an Open Source environment to model vertical web application layers and to generate the artifacts of these layers on specific vertical web frameworks.

Modeling allows capturing the complete sets of functional needs of an artifact for the benefits of one single or many web applications during all their life-cycle.

Generation allows producing the executable code and configuration of an artifact for the concerned applications on one single or many targeted web frameworks successively during all the applications' life-cycle.

SIDE is composed of 6 main products:

- A set of Meta-models: they describe a precise layer of the targeted web applications. For instance, the data meta-model describes the content types and their association which may be declined in an ECM application.
- A set of Modelers: a SIDE modeler is an Eclipse plug-in or a web application enabling the graphical modeling of artifacts of targeted functional layers through the production of models. It exists at least one modeler per meta-model and a modeler is conform to a single meta-model.
- A set of Generators: a SIDE generator is a Model to Model and Model to Text Java-based program enabling the transformation of the models on another models for refinement or on a specific web framework for application instantiation. A generator processes a set of models of a single meta-model.
- A set of Deployers: a SIDE deployer is a Java-based component which produces the packaging of the generating artifacts and deploys it on the targeted framework. For instance, the Alfresco deployer produces war files from the generated AMP files. A deployer is associated to a specific framework and is able to package generated artifacts relative to various meta-models.
- A set of advanced server components: In order to extend some features of the targeted frameworks, SIDE provides a set of modules which allows benefiting of very practical and intuitive features associated to the generated artifacts of designed models. These modules are provided by some generators and are deployed on requests by the concerned deployer. For instance, the facet navigation module allows integrating facet navigation on Alfresco.
- A build and extension module development environment: Under SIDE, you can benefit from a maven environment to develop your framework extensions in java, javascript, freemarker, xml, ... and an integrated build process which packages and deploys the complete sets of deliverables (generated components, advanced components and your own extension modules

SIDE brings the following benefits to web projects:

- It ensures the sustainability of the web application through the achieved model which can be generated on successive releases of one web framework.
- It ensures the sustainability of the web application through the achieved model which can be migrated on other more adapted or more effective frameworks at the time being.
- It ensures the delivery of qualified web components
- It ensures the build and integration of this components in the targeted web frameworks
- It ensures the delivery of best of breed web components as the BlueXML team has the knowledge and makes use of the more sophisticated and effective capabilities of the targeted web framework.



- It provides advanced features of the generated web components like table with advanced search capabilities or forms with creation and editions of “many to many” associations.
- It provides the integration procedures and components to support complex architecture deployment like SSO.

Multiple years of Research and development conduce to stabilize and to qualify an impressive set of the following products of the SIDE suite:

- “SIDE Requirements” allows:
 - modeling Web Application Requirements as artifact,
 - transforming this Requirements into Entity/Relation, Portal structure and Workflow models.
 - generating RFQ-based documents
- “SIDE Data” allows:
 - modeling Application Data structure through definition of objects, attributes and associations as artifact,
 - generating them on web application framework like Alfresco and Hibernate.
- “SIDE Workflow” allows:
 - modeling business processes as artifact,
 - generating them on web application framework like Alfresco through JBPM.
- “SIDE Portal” allows:
 - modeling Web Portal structures as artifact,
 - generating them on portal engines like Liferay, Alfresco Share and Jboss Portal.
- “SIDE View” allows:
 - modeling web object display as artifact,
 - generating views on web framework like Alfresco, dojo,
- “SIDE Form” allows:
 - modeling web object edition and searching as artifact,
 - generating forms on form engines like Chiba (Xforms) or on Alfresco.

Technological overview

SIDE is based on the MDA (Model Driven Architecture) to model and to generate web application components.

Currently, SIDE generates these artifacts on the following frameworks:

- ECM Alfresco 2.x and 3.x,
- Chiba 1.x Xforms engine,
- Liferay 5.x portal.

The SIDE modeling environment and the generators are mainly based on Eclipse and the MDA standard libraries like Topcased, EMF and ATL.

SIDE also provides some advanced server components on Alfresco to extend some features like Share views, SQL access to the generated content types.

The MDA approach on which is based SIDE suggests to model functional layers using DSL – Domain Specific language – languages able to capture the functional invariant in models and able to generate these models on specific frameworks: this allows to make perennial a part of the application through models which offers a longer life cycle than the code of the frameworks, always outdated.

BlueXML chooses to apply these techniques to the Alfresco ECM frameworks by providing meta-model based on requirements, content types, workflow, portal structure, form and view models. For each of these meta-model, SIDE provides inter-models transformation, generators and deployers for various versions of Alfresco and of specialized frameworks like jBPM for workflow, Chiba for form, Alfresco Share and Liferay for portal structures.

The community www.side-labs.org site is the entry point for those who want to contribute to or to learn more on the MDA approach applied to Alfresco and other frameworks. It allows to download the open source community version of SIDE.

The corporate site www.bluxml.com is the entry point for those who want to get professional services on top of SIDE.



SIDE Documentation Structure

This guide is a part of the SIDE documentation suite. The available guides are:

- SIDE Installation Guide describes the installation process of the full set of the SIDE products.
- SIDE Concept and Starting User Guide – provides an overview of the features and capabilities of SIDE and of the Model Driven Engineering concepts which SIDE relies on. It describes the general architecture, the main components and how to start to optimize your work.
- The Web Components Generation guides:
 - The SIDE Requirement Modeling User Guide describe the modeling of requirements and the successive transformations to make the good start and to drive web applications
 - The SIDE Data Modeling User Guide describes the modeling of the data structure of the application and the successive transformations to generate data persistence.
 - The SIDE Workflow Modeling User Guide describes the modeling and generation of Web Workflow on best-of-breed Workflow engines.
 - The SIDE Web Forms Modeling User Guide describes the modeling and generation of Advanced Web Form components for Web Applications.
 - The SIDE Web Views Modeling User Guide describes the modeling and generation of Advanced Web View components for Web Applications.
 - The SIDE Portal Modeling User Guide describes the modeling and generation of Web Portal structure on best-of-breed Web Portal Framework.

Typographic conventions

In all the SIDE User guides, the following conventions are used:

Format	Description
<i>Italic</i>	Used when referring to items (menu items, buttons, list items) of the UI
<i>Bold Italic</i>	Used when referring to text that you type yourself
CAPITALS	Used to indicate keys of the keyboard
<i>File -> New</i>	Used to indicate a menu items sequence. This means: Fig_1. On the menu bar, click 'File' Fig_2. Then click 'New'
<code><config>/alfresco</code>	The courier font is used for path. Forward slashes are used (if you deploy on windows do not forget to convert using backward slashes and drive prefix) Path parameters (like config) are given between '<' and '>'. Upon deployment, replace them by your installation value without '<' and '>'.

Predefined Parameters

In all the SIDE User guides, the following parameters have the following meaning:

Parameters	Meaning
------------	---------



<eclipse_home>	The path to the Eclipse installation directory: this directory contains at least the directories configuration, plugins, features, dropins.
<alfresco_home>	The path to the Alfresco installation directory: this directory contains at least the directories bin, extras, amps, licenses, tomcat
<alfresco_webapp>	The path to the Alfresco webapp: this directory is labeled 'alfresco' and contains at least the directories META-INF, WEB-INF
<alfresco_share_webapp>	The path to the share webapp: this directory is labeled 'share' and contains at least the directories META-INF, WEB-INF
<xforms_webapp>	The path to the xforms (either Chiba or Orbeon) webapp: this directory is usually labeled 'xforms' and contains at least the directories forms, META-INF, WEB-INF
<liferay_home>	The path to the Liferay installation directory: this directory contains at least the directories tomcat, extras
<liferay_webapp>	The path to the Liferay webapp: this directory is labeled liferay and contains at least the directories META-INF, WEB-INF
<server_adr>	The web address of the server where you deploy the targeted application. For instance, if you deploy Alfresco on the host 'myHost' and the port '8080', server_adr is 'myHost:8080'.
<app_server_path>	The path to the application server where you deploy the targeted application. For instance, if you deploy Alfresco under the directory '/opt/alfresco' using Tomcat, app_server_path is '/opt/alfresco/tomcat'.

About This Manual

This book assumes that you have a good knowledge of the Linux or Windows operating system on which you are installing the product

The steps relative to the integration with a targeted framework are performed using Alfresco, the leading open source ECM. For this part, this book assumes that you have a good knowledge or an access to the documentation of this web framework.

This book, the Installation guide, is intended to help you:

- Install the SIDE components.
- Configure the SIDE components.
- Install and Configure a targeted web framework.
- Learn the major steps of integration on this framework.

This manual is divided in 5 main chapters:

- The 'SIDE Product Architecture' chapter describes the software architecture of SIDE for the modeling of applications and for the deployment of applications.
- The 'SIDE Installation Procedure' chapter describes the installation steps of SIDE.
- The 'SIDE Installation test' chapter describes a way to test the correct installation of SIDE through a simple MDA procedure.
- The 'SIDE Feedback procedure' chapter explains how you can help us to improve SIDE with usage tracking of SIDE.
- The 'Targeted Frameworks Installation procedure' does not intend to explain the installation procedure of each frameworks targeted by the SIDE generators and deployers but to give useful information related to the installation and configuration of these frameworks.



SIDE PRODUCT ARCHITECTURE

SIDE is a MDA development environment which allows generating models on other models or on web frameworks.

It is mainly based on Eclipse to access:

- the modelers : graphical environment to build models,
- the generators : MDA transformation Eclipse plug-in to produce the targeted framework components,
- the deployers : Eclipse plug-in to package and to deploy the targeted framework components in installed web framework instances.
- the Maven projects space where web frameworks extensions may be developed and built.

The users interacts with:

- the various modelers through the modeler windows,

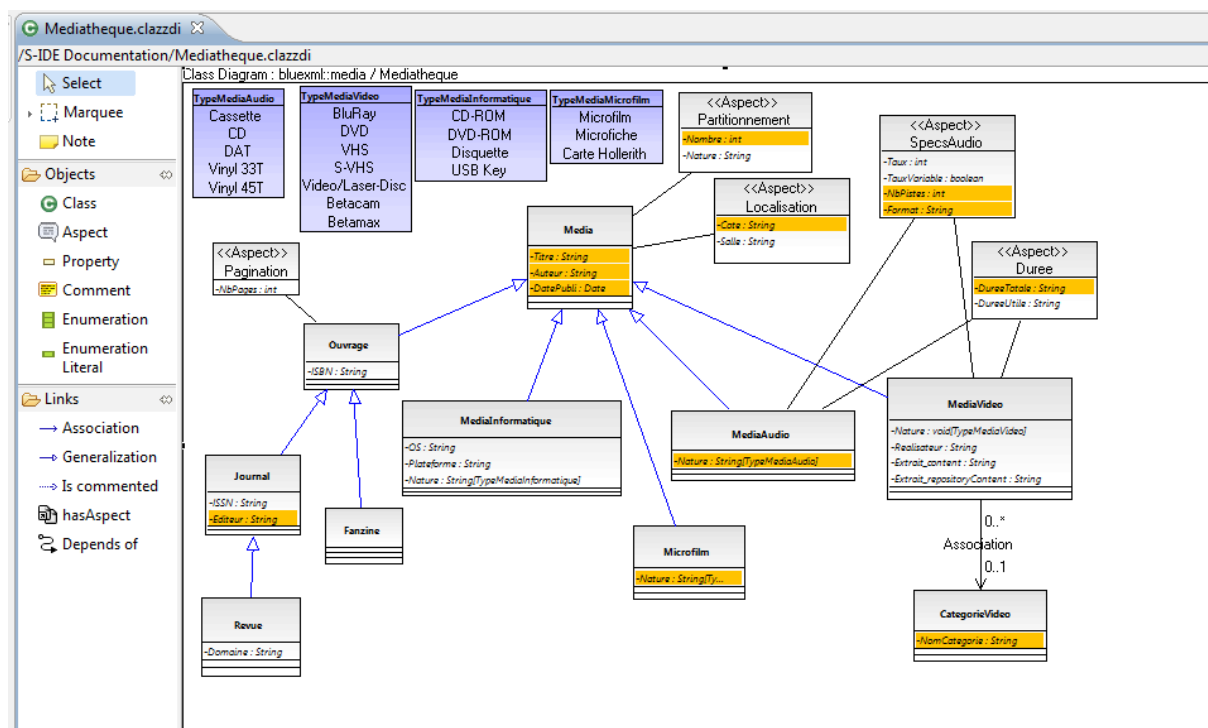


Illustration 1: Example of modeler: the data modeler

- the generators and the deployers through a specific window under Eclipse which is a guided modeler of an application model.



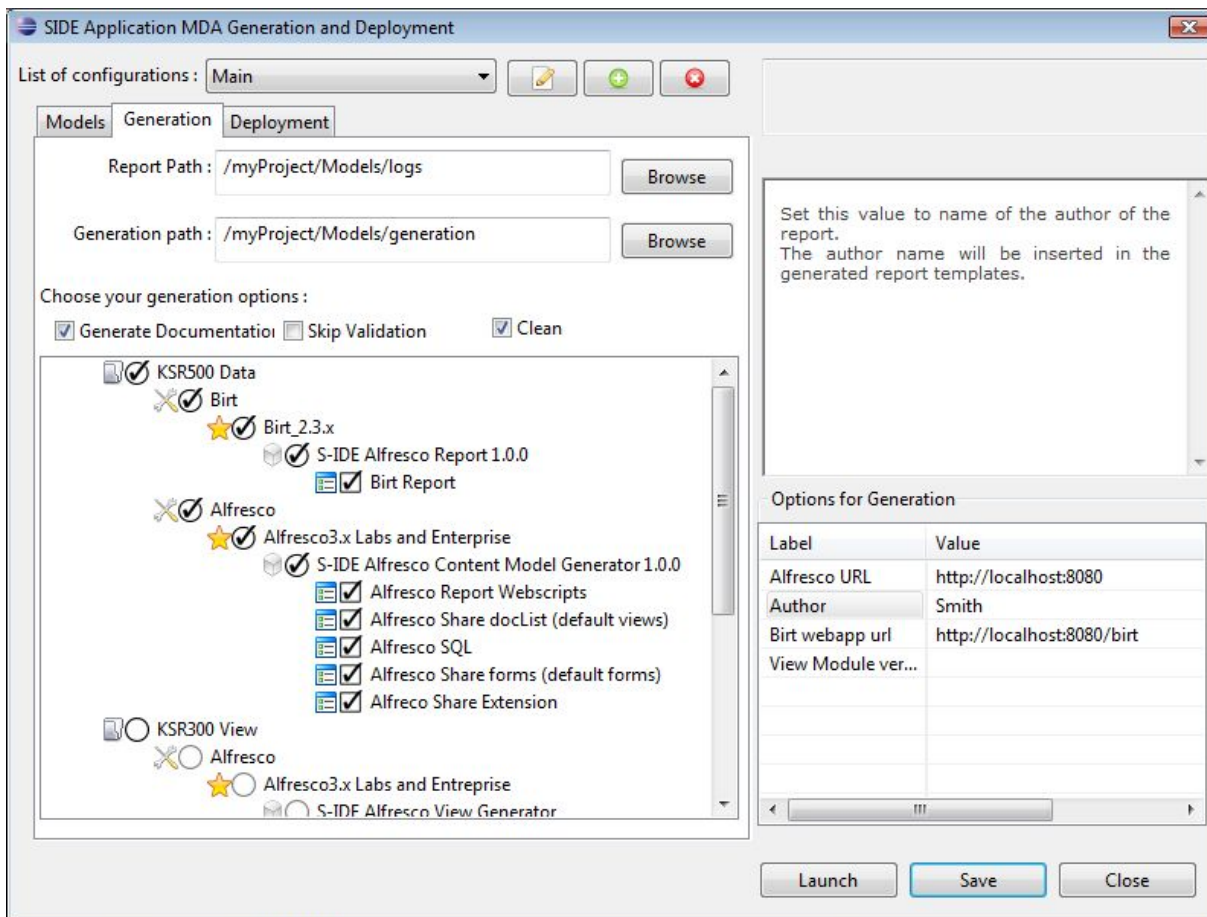


Illustration 2: The application modeler window

So, the process of installation and configuration of the main parts of SIDE are mainly through Eclipse mechanisms.

SIDE also comes with some server modules known as SIDE Advanced components which are useful to extend the capabilities of the targeted frameworks.

For instance, SIDE provides a facet navigation library for Alfresco and a module to publish the Alfresco schema through a relational schema which may then be requested using SQL.

These server modules are provided at generation with the generated artifacts and are deployed on the targeted installed frameworks by the deployers.

The following figure gives an overview of the product's architecture:



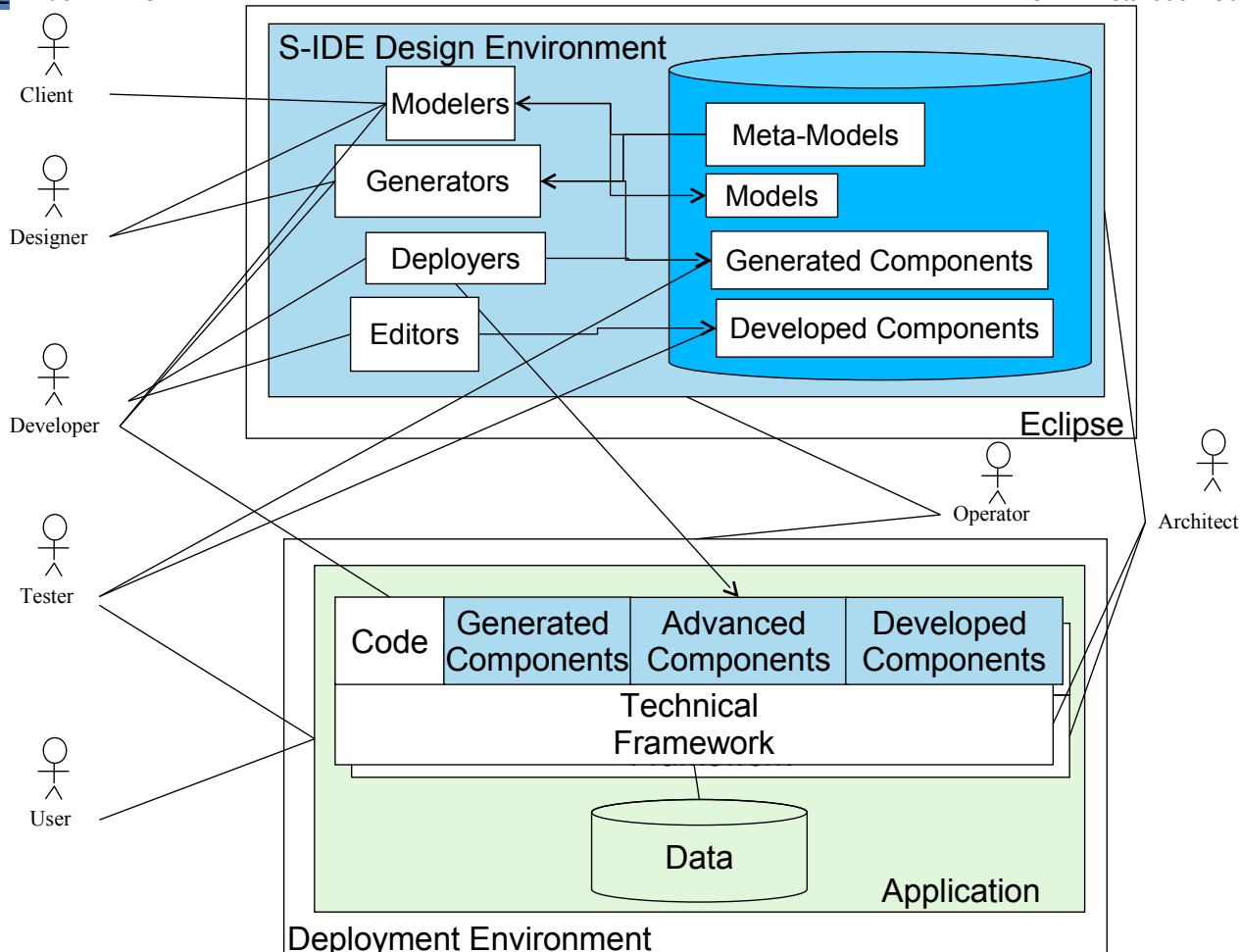


Illustration 3: SIDE Development and component server architecture

The generated advanced and developed components are deployed on the targeted framework instances by the deployers, a deployer being linked to a framework.

The installation procedure described hereafter describes the installation of the SIDE Design environment.

As we also dealt with the development of the deployers and with the deployment of complex architecture, especially on Alfresco, Share, Liferay and Chiba on several customer projects, we developed a fair amount of expertise in these domains. Feel free to contact us or to use our wiki and forum to get assistance on the deployment.

On this figure, distinction is made between the Design environment where SIDE components have been installed and the Deployment environment where SIDE is not installed except for the Advanced components which are extension libraries provided by SIDE to extend the capabilities of the targeted framework and which are free of use.



SIDE INSTALLATION PROCEDURE

The SIDE Installation process implies the successive installation of various components in the Eclipse environment. These steps are described successively in the following chapters:

- The first chapter describes the installation of the Eclipse environment.
- The second chapter describes the installation of SIDE and of the components the SIDE plugins are dependent on.
- The last chapter gives a check-list to perform after installation.

Java Installation

Step 1: Download and install JDK 1.5/1.6

You need the JDK 1.5/1.6 on your machine to be able to run SIDE.

Note: as SIDE compiles, packages through tools like javac, jar, ant and maven, the JRE cannot be used to run SIDE.

For Windows Platform

Download the Java Development Kit from:

- http://java.sun.com/javase/downloads/index_jdk5.jsp

Then execute the following steps:

1. Save the setup file `jdk-1_5_0_16-windows-i586-p.exe` on your system,
2. Run the file
3. Follow the instructions.

For Linux Platforms

Download the Java Development Kit from:

- http://java.sun.com/javase/downloads/index_jdk5.jsp

Then execute the following steps:

1. Save the setup file `jdk-1_5_0_16-linux-i586.bin` on your system.
2. Open a terminal window
3. Go to the directory where you had download the JDK
4. Execute the following command:
 - `sh ./jdk-1_5_0_16-linux-i586.bin`
5. follow the instructions.

Note: SIDE is only built on 32-bits architectures using 32-bits JVM. If you want to deploy SIDE on a 64-bits architecture, use a 32-bits JDK.

Step 2: Configure JDK 1.5/1.6

You must now set up the system environment variables to JRE 1.5 so that Java programs know where to find standard system files they would need.

On Windows Platform

- Open the system properties dialog using Windows key + Pause or choose System in the Control Panel.
- Click on the environment variables button. In the system variables section, look for JAVA_HOME and JRE_HOME.
- If not present, you need to add them, according the paths you chose to install each of the JDK and JRE:



- Set JAVA_HOME to C:\Program Files\Java\jdk1.5.0_16
- Set JRE_HOME to C:\Program Files\Java\jre1.5.0_16 or

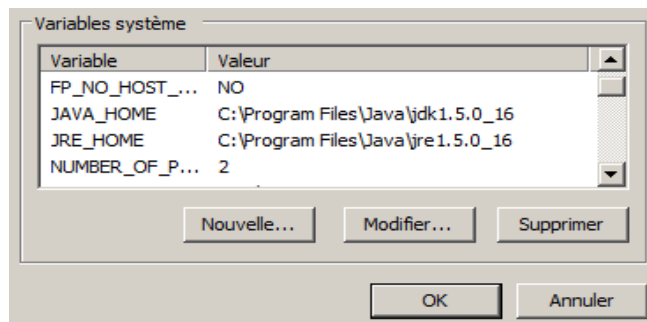


Illustration 4: Java environment variables

On *n*x Platform

- Export the variable in the login script of the shell you use like the following:
 - export JAVA_HOME=the directory where java is installed

Note: jdk 1.5.0.16 and 1.6.0.07 have been tested successfully with SIDE.

Eclipse Installation

SIDE comes as an Eclipse RCP which integrate the modelling package of Eclipse Galileo including EMF, Topcased and Acceleo.

Step 1: Download SIDE

Go to <http://www.side-labs.org/wiki/index.php/Documentation:Download:Overview> site to download SIDE:

Versions on Windows, Linux and MacOS are available.

You will get a compressed file you must unzip, for instance, under 'c:\Eclipse' for windows or '/opt/eclipse' for Linux.

In the following, your Eclipse home directory is labeled '<Eclipse Home Directory>'.

Step 2: Configure SIDE

There is no so much configuration steps to perform as SIDE will run as is.

Nevertheless, you can configure the file '<Eclipse Home Directory>/side.ini' in order to:

- change the language by adding the following lines :


```
-nl                                     <code                                lang>
```

Note: you may also create a shortcut "side.exe -nl <code lang>"; for instance "side.exe -nl en" to start SIDE in english.
- improve the performance or if you experience memory leak problem by adding the following lines:

```
--launcher.XXMaxPermSize
256m
-vmargs
-vm
"<path to your jdk bin folder>"
-Xms64m
-Xmx384m
```

Adjust the values to your environment

- to avoid to open the SIDE Web page at startup:



```
-nostartupbrowser
```

- a bug has been identified on the "Open With" command which doesn't work well if the launcher.DefaultAction is not present; so, if you experience this problem, add:

```
--launcher.defaultAction  
openFile
```

- On windows XP/Vista/7, according to the level of security, it seems that the SIDE installation failed to copy some maven libraries in your local ".m2" maven repositories. You will experience this problem when you generate artifacts through the SIDE MDA Generation process:

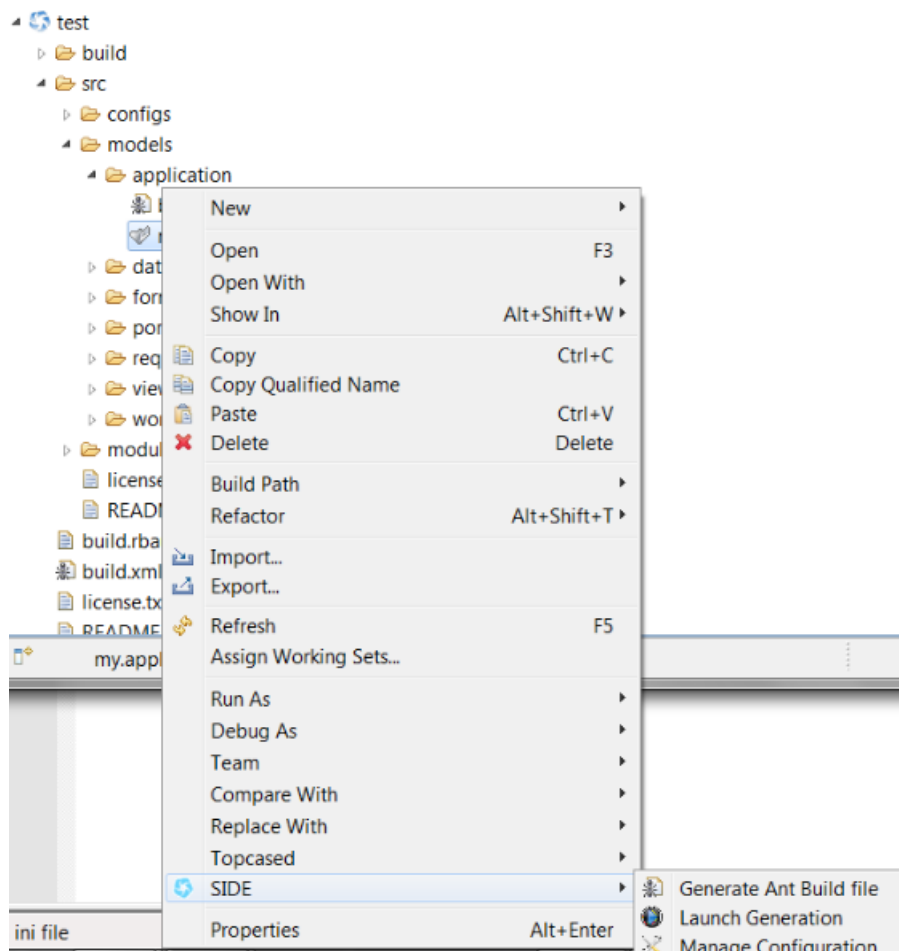


Illustration 5: SIDE MDA Generation commands

Then you have access to the window to start the generation.

Let's suppose you run the MDA generation and you get an error window indicating that required artifacts are missing :



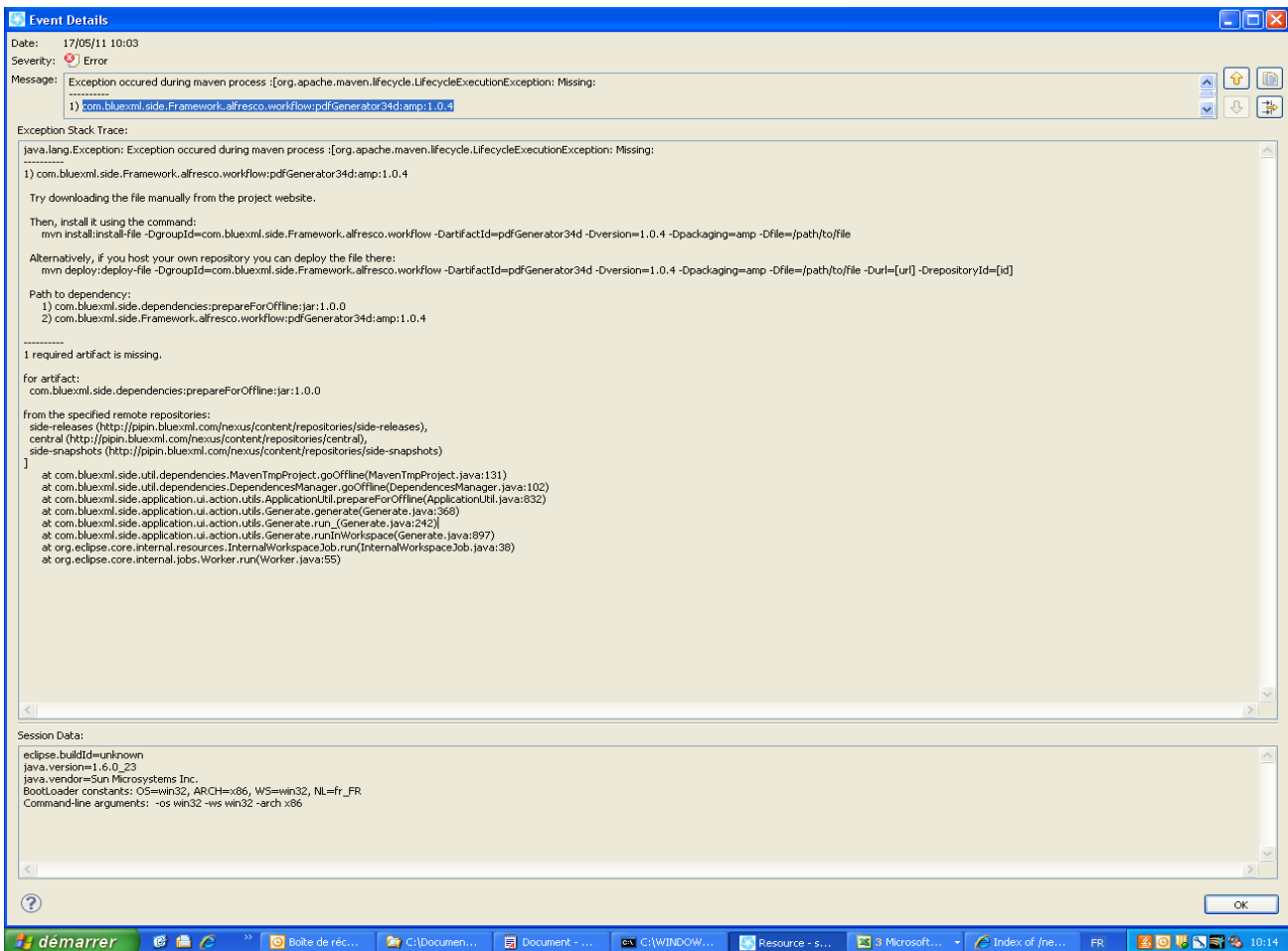


Illustration 6: Error on missing required artifacts

In this case, try one of the following approaches:

- Under SIDE, Select "Windows->Preferences->SIDE->SIDE Maven Configuration" and click on "install". If you want to use a local maven repository exclusively for SIDE, change the "Local Repository" path which is by default set to the default maven local repository path for the current user.

- if you only use maven and your ".m2" local repository for SIDE, suppress completely the ".m2" folder and restart SIDE.

- if you use maven and your ".m2" local repository for other products:

- . copy the '<Eclipse Home Directory>/plugins/com.bluxml.side.Util.dependencies.repository.enterprise_<V.0.vv.vxx>.jar' under /tmp
- . extract the content of the jar (use 'jar xvf' command or izarc tools or rename in .zip)
- . extract the content of the file tmp/com.bluxml.side.Util.dependencies.repository.enterprise_<V.0.vv.vxx>/com/bluxml/side/util/dependencies/repository/enterprise/mavenRepository/m2repositoryForSIDE.zip under '<user home>/m2/repository'

Step 3: Start SIDE

click on the '<SIDE Home Directory>/side.exe' on windows or 'side' on linux|MacOS.



Note: If you experience trouble starting SIDE on a 64-bits architecture, look in the '<SIDE Home Directory>/workspace/.metadata/.log' file to identify the problem: it is usually related to a missing library of a package you have to install.

Checklist

Go to Window -> Preferences.

Check that the entry SIDE is present, with the following subentries:

- Feedback
- Term of Use

Go to Help > Install New Software and click at the bottom on the right on 'already installed'. You should have lots of SIDE entries, stored under BlueXML provider name.



SIDE INSTALLATION TEST

To continue the check list, we will now open the SIDE modelers to test them:

- In Eclipse, go to Window -> Open Perspective -> Other.
- Check that 'SIDE Modeling' is present.
- Select it and check that the perspective is accessible by clicking on OK.
- The following figure should be visible.

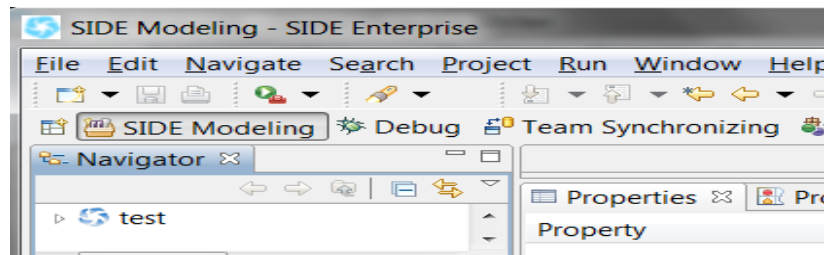


Illustration 7: SIDE modeling perspective

Select File -> New -> Project

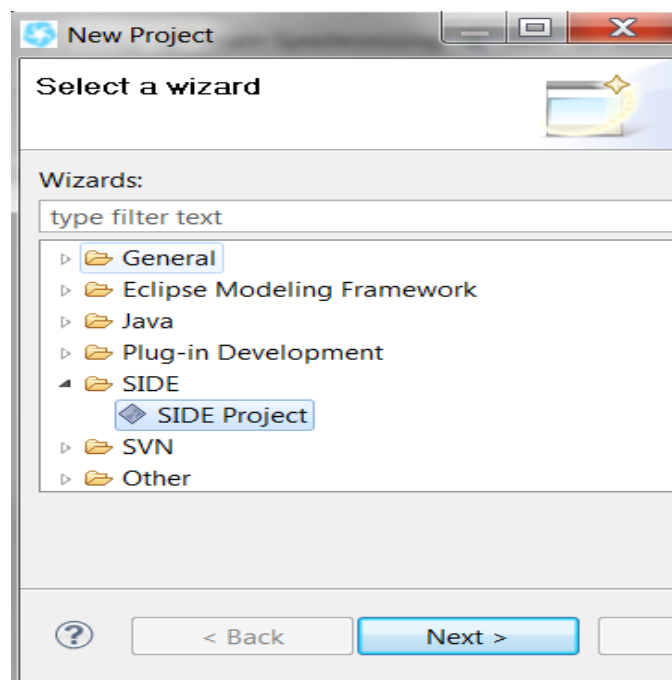


Illustration 8: SIDE Project selection

Select 'SIDE project' under 'SIDE' category.

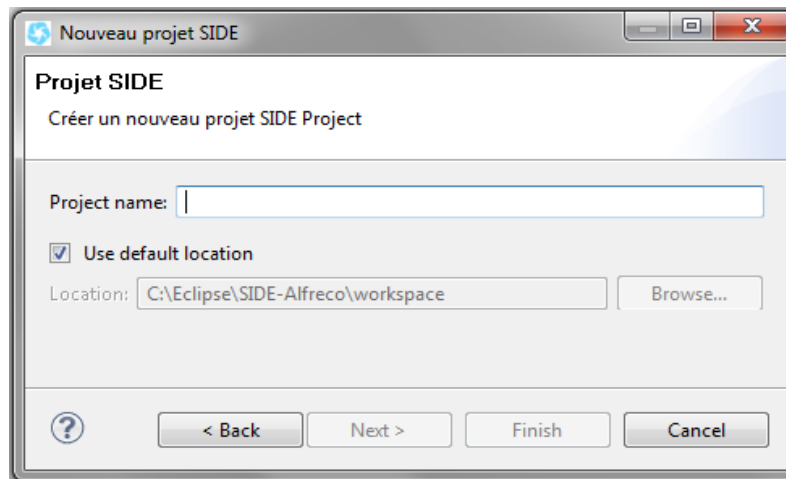


Illustration 9: Application's Project creation

Give a name to your project, for instance: test and click 'next'.

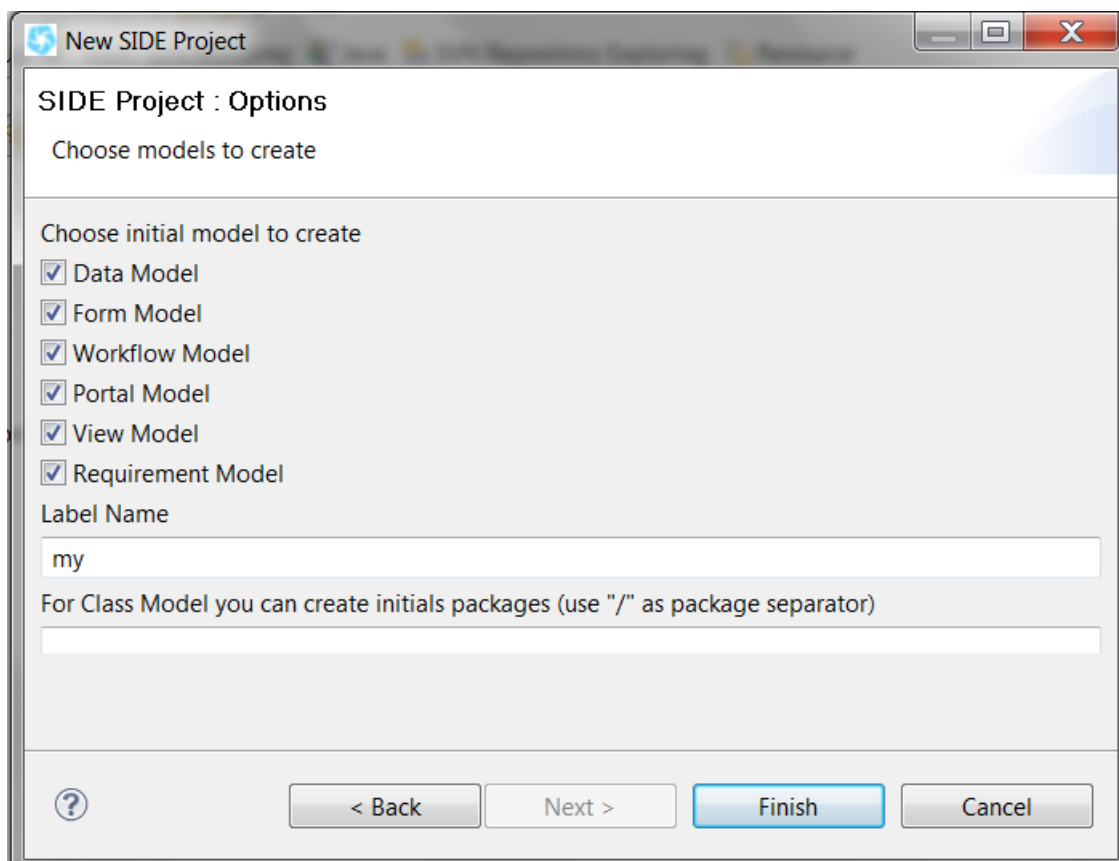


Illustration 10: Models creation

Click on the checkbox to automatically create models and click 'Finish'

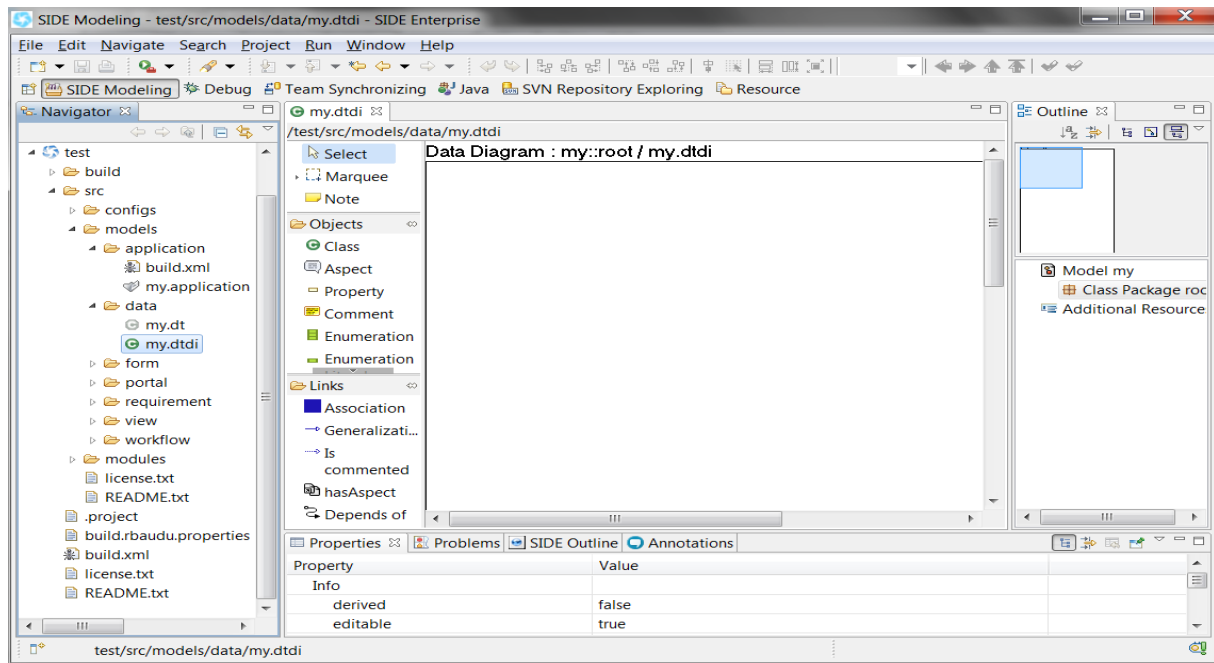


Illustration 11: Application's Project in your workspace

Your project has been created in your workspace.

If you want to create other MDA models:

- Select File -> New -> Other

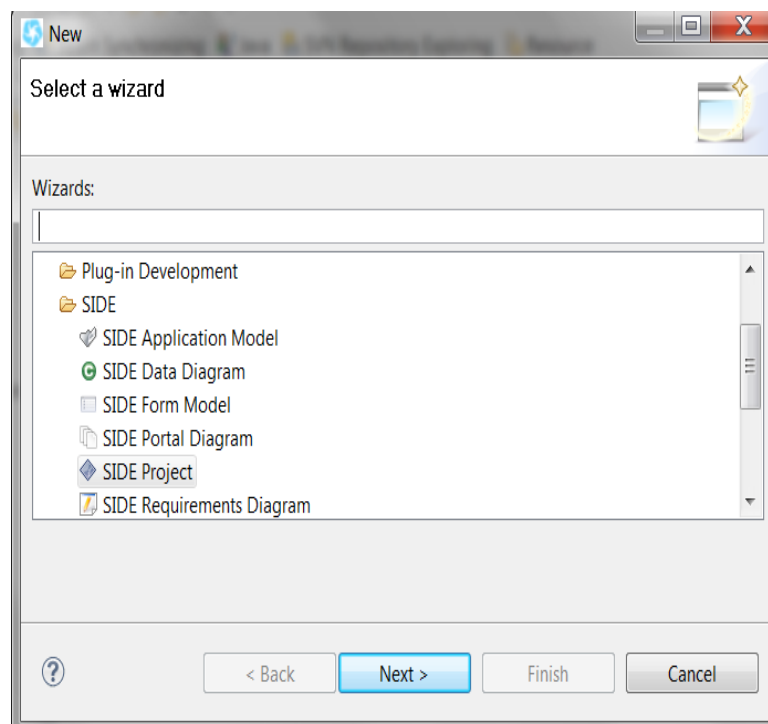


Illustration 12: New Models Window

- Browse in the list to find 'SIDE'



- Select for instance SIDE Data Diagram, Click 'Next', Enter the name of your model, for instance mediaLibrary and click 'Finish'.

Two files are created in your project:

- '*mediaLibrary.dt*' is the data model containing all the designed data structure.
- '*mediaLibrary.dtd*' is synchronized with the data model and contains the designed data structure complemented with the graphical information necessary to position the data elements in the SIDE Data modeler.

The '*mediaLibrary.dtd*' model is automatically opened in the SIDE Data modeler:

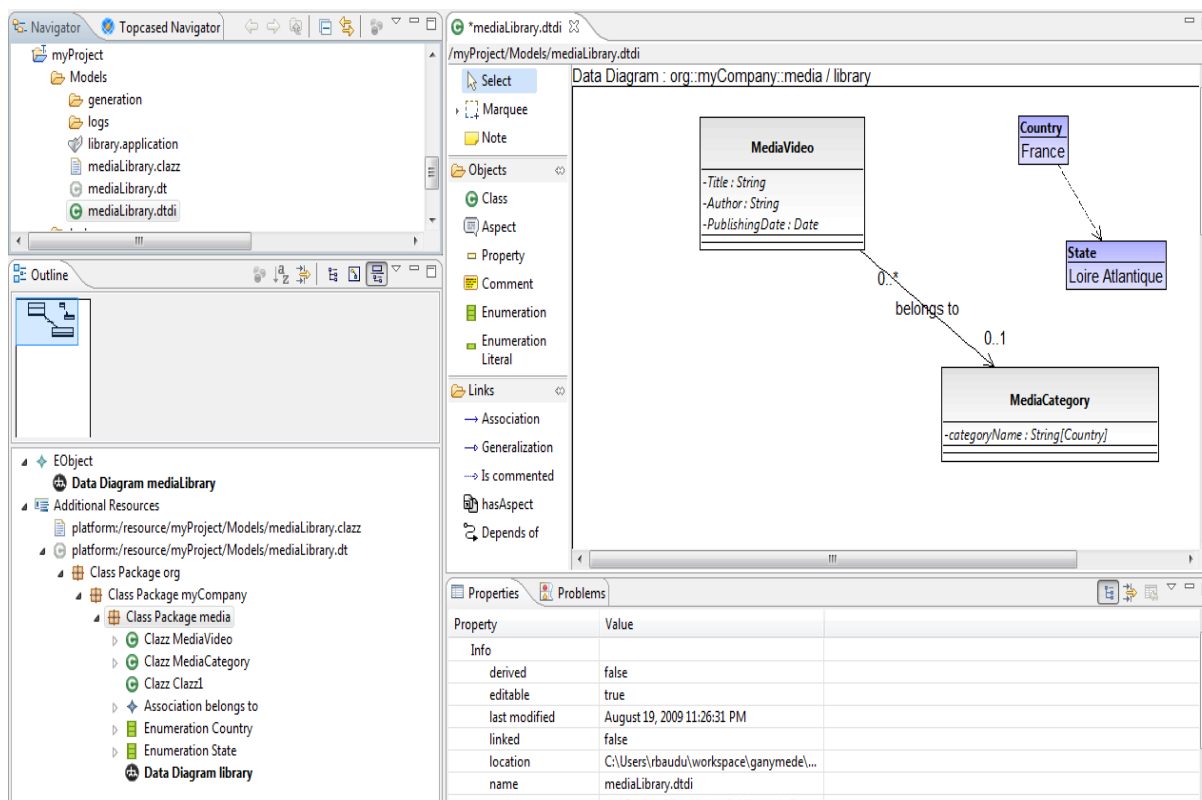


Illustration 13: SIDE Data Modeler

You can just create a new type of content by clicking on 'Class' and dragging it in the right panel. By default, the class is labeled 'clazz1': double-click on it to change its name.

The MDA process uses Maven repositories to resolve the dependencies to used libraries. To feed your local repository, you need to launch at least one generation.

To perform this, under the "models/application", right-click on the "my.application" model file and select "SIDE->Manage Configuration".

The Application MDA Configuration window is opened :



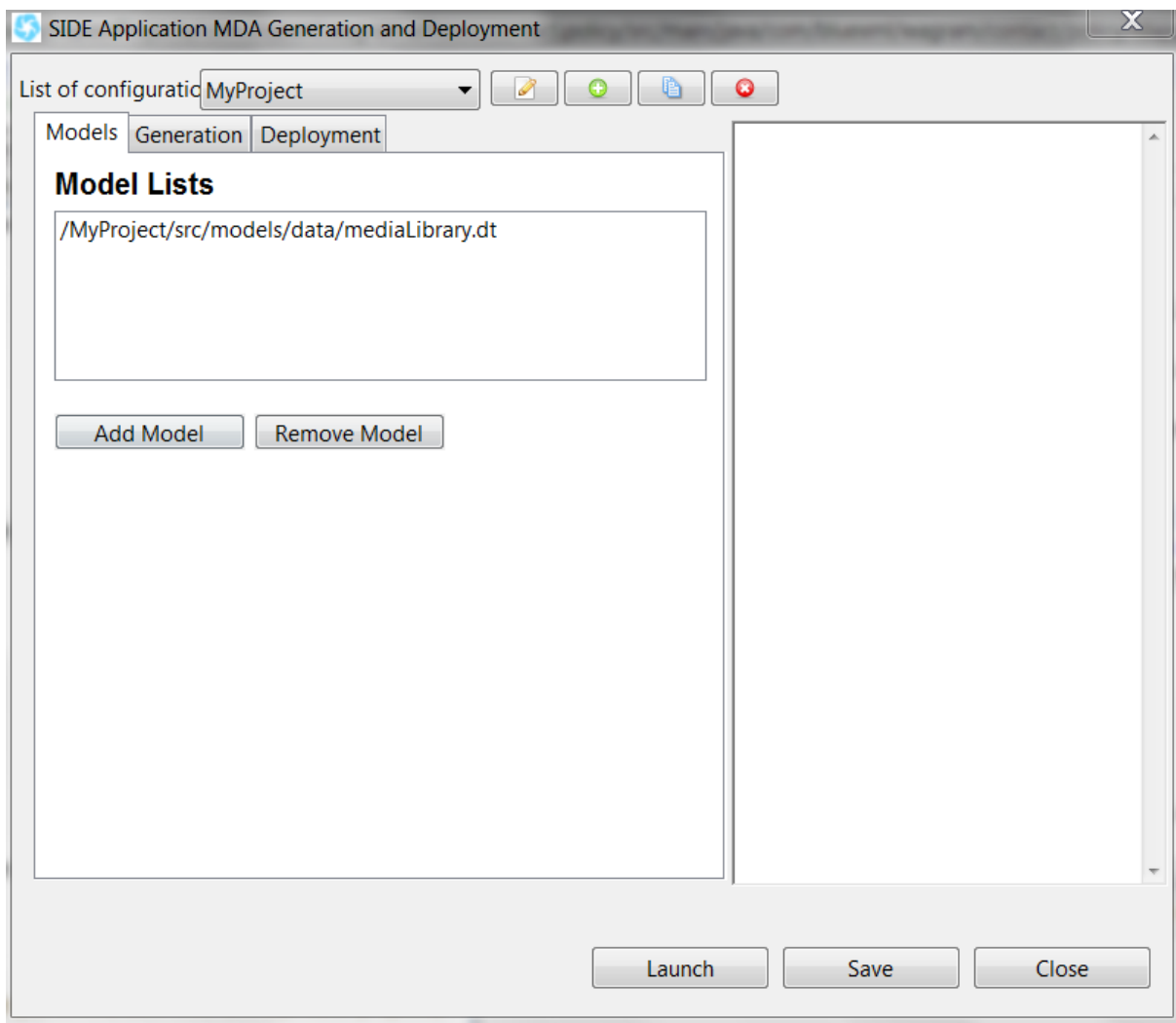


Illustration 14: Application MDA Manage Configuration Window

Add your MediaLibrary Data model using the "Add Model" button.

Then click on the "Generation tab"

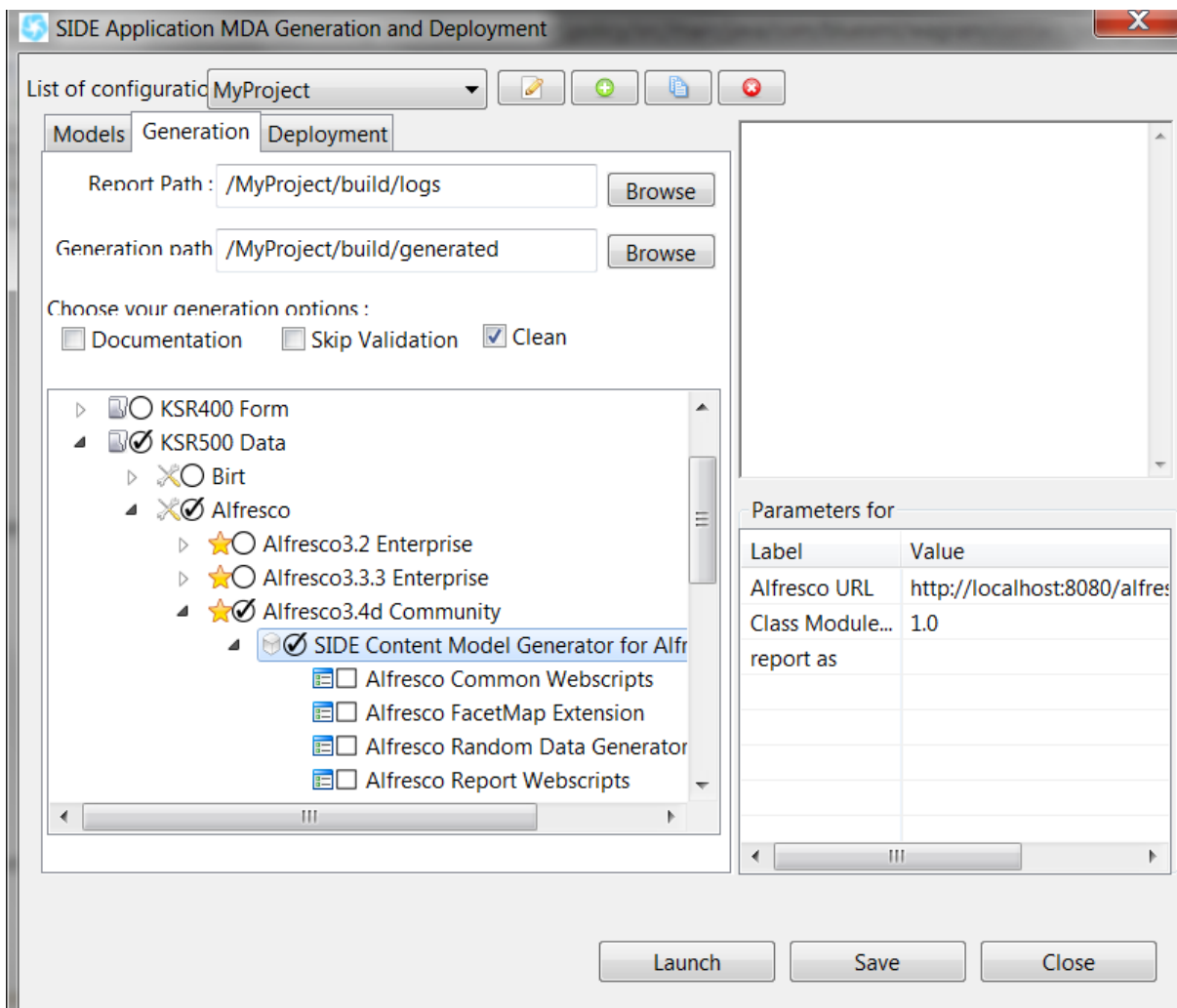


Illustration 15: Generation tab of the MDA process

Select the metamodel "Data" and the generator corresponding to your Alfresco version.

Click on Save and then Launch to start the generation of the Alfresco content type extensions.

This first generation will update your local maven repository with the necessary dependant libraries.

Note:

- *You can force the update of your local maven repository using the "Window -> Preferences -> SIDE -> SIDE Maven Configuration" window by clicking on "Install".*
- *This command is also useful if you experienced dependencies trouble (maven artifact not found) during generation on windows to force the installation of the maven dependant libraries. It seems that right management on Windows may prevent the installation of the maven dependencies at generation; the "install" button may bypass this problem.*
- *If you want to have a specific for the local maven repository used by SIDE, change the path given in the property "Local Repository". The default value is the standard maven local repository path for the current user, which means that this default path is shared by SIDE and other applications using Maven.*



SIDE INSTALLATION EXTENSIONS

SIDE comes with MDA tools but also provides an environment to develop and build maven-based java/javascript/xml/ftl projects.

Installation of plug-ins

If you want to plainly use this environment, develop with these languages and build using maven process, follows the procedure hereafter to install the necessary tooling:

- In Eclipse, go to Help -> Install New Software -> Add
- Enter under location : <http://download.eclipse.org/releases/galileo>
- Click on OK
- The following figure should be visible

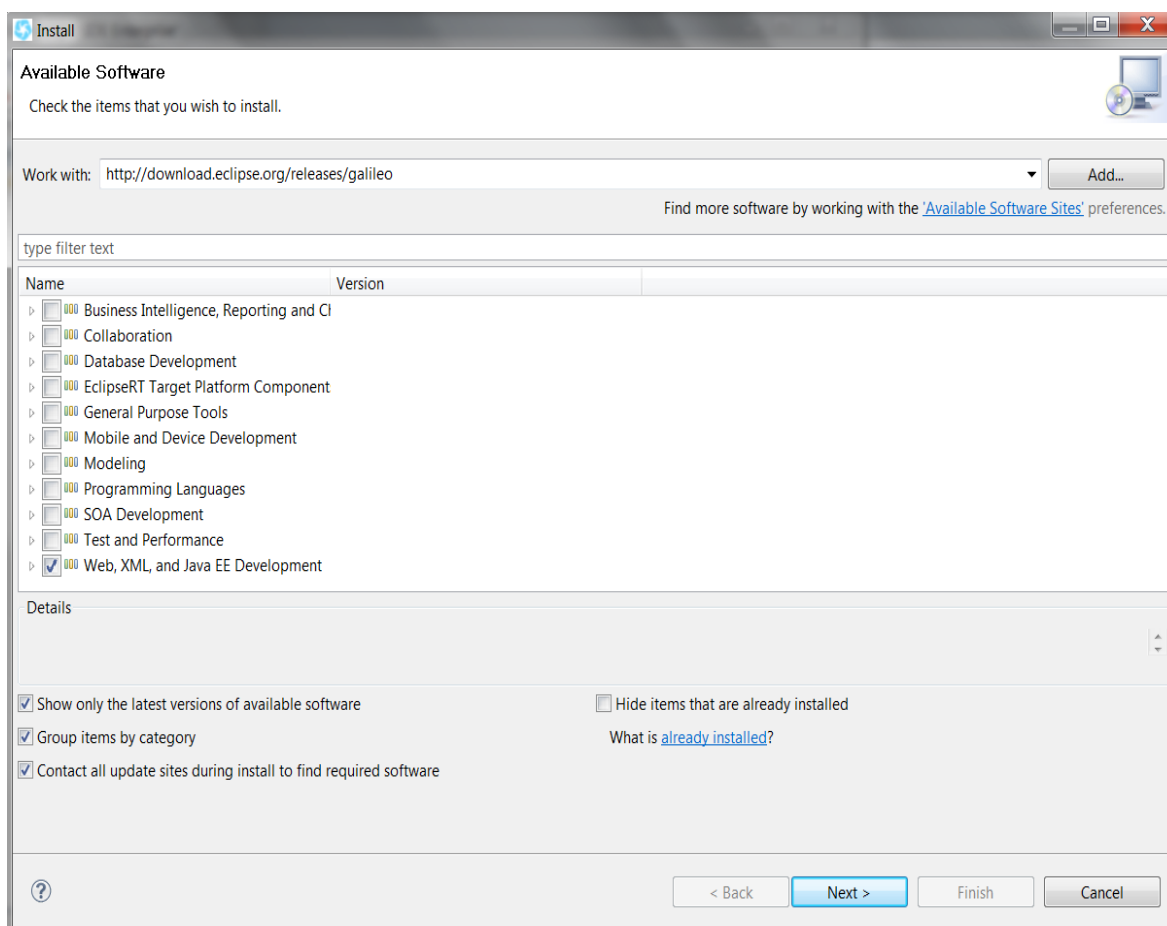


Illustration 16: Galileo update site

- Click on the arrow before "Web XML and Java EE Development"
- Select WST Server Adapter (required for Maven Integration) and the tools you want to use. For instance:
- Click on Next -> Finish.
















<input checked="" type="checkbox"/>		Eclipse Java EE Developer Tools	3.1.1.v200908101600-7_7EGrjFQRwRb4P511e...
<input checked="" type="checkbox"/>		Eclipse Web Developer Tools	3.1.1.v200908120400-7R77FStEVw2z07WtDz-...
<input checked="" type="checkbox"/>		Eclipse XML Editors and Tools	3.1.1.v200907161031-7H6FMbDxtkMs9OeLGF...
<input checked="" type="checkbox"/>		Eclipse XSL Developer Tools	1.0.2.v200908271520-7R7T8sFlhIehWg454rq_...
<input checked="" type="checkbox"/>		JavaScript Developer Tools	1.1.2.v200908101420-77-FGDCcNBDjBXMObb...
<input type="checkbox"/>		JST Server Adapters	3.1.1.v20090813-777HFDPCcNBBz0BjJU4BA8
<input type="checkbox"/>		JST Server Adapters	3.1.0.v20090311c-20-7w311A1422c
<input type="checkbox"/>		JST Server UI	3.1.1.v20090817-7A2FBb9xFaHovXGLUCTMQ2...
<input type="checkbox"/>		JST Web UI	3.1.1.v200908121609-7E77FBiDlwYcICNdz-5z-...
<input checked="" type="checkbox"/>		PHP Development Tools (PDT) SDK	2.1.1.v20090707-1108-51584QACJRCUeNRk...
<input checked="" type="checkbox"/>		Rich Ajax Platform SDK (RAP)	1.2.2.20100126-1129
<input checked="" type="checkbox"/>		Web Page Editor (Optional)	2.2.2.v201001271707-43799oB5855B7A2242
<input checked="" type="checkbox"/>		WST Server Adapters	3.1.0.v200906020451-51DFAkF77g7GCNc

Illustration 17: Maven and development tooling to select

- Accept license requirements and restart when needed

The next step is to install the Maven plugin:

- In Eclipse, go to Help -> Install New Software -> Add
- Enter under location : <http://m2eclipse.sonatype.org/sites/m2e>
- Click on OK
- Select "Maven Integration" and click OK

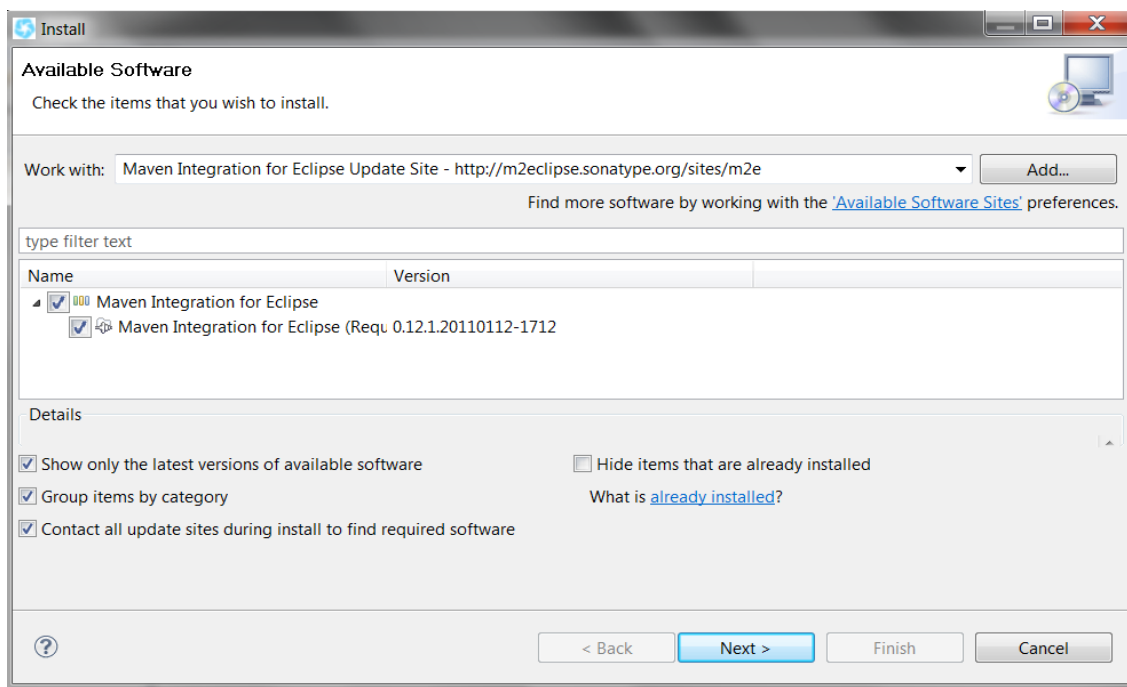


Illustration 18: Maven Plugin installation

- Accept license requirements and restart when needed

The next step is to install the Freemarker IDE:



- In Eclipse, go to Help -> Install New Software -> Add
- Enter under location : <http://download.jboss.org/jbosstools/updates/JBossTools-3.1.1.GA/>
- Click on OK
- Select "All JBoss Tools -> FreeMarker IDE" and click OK

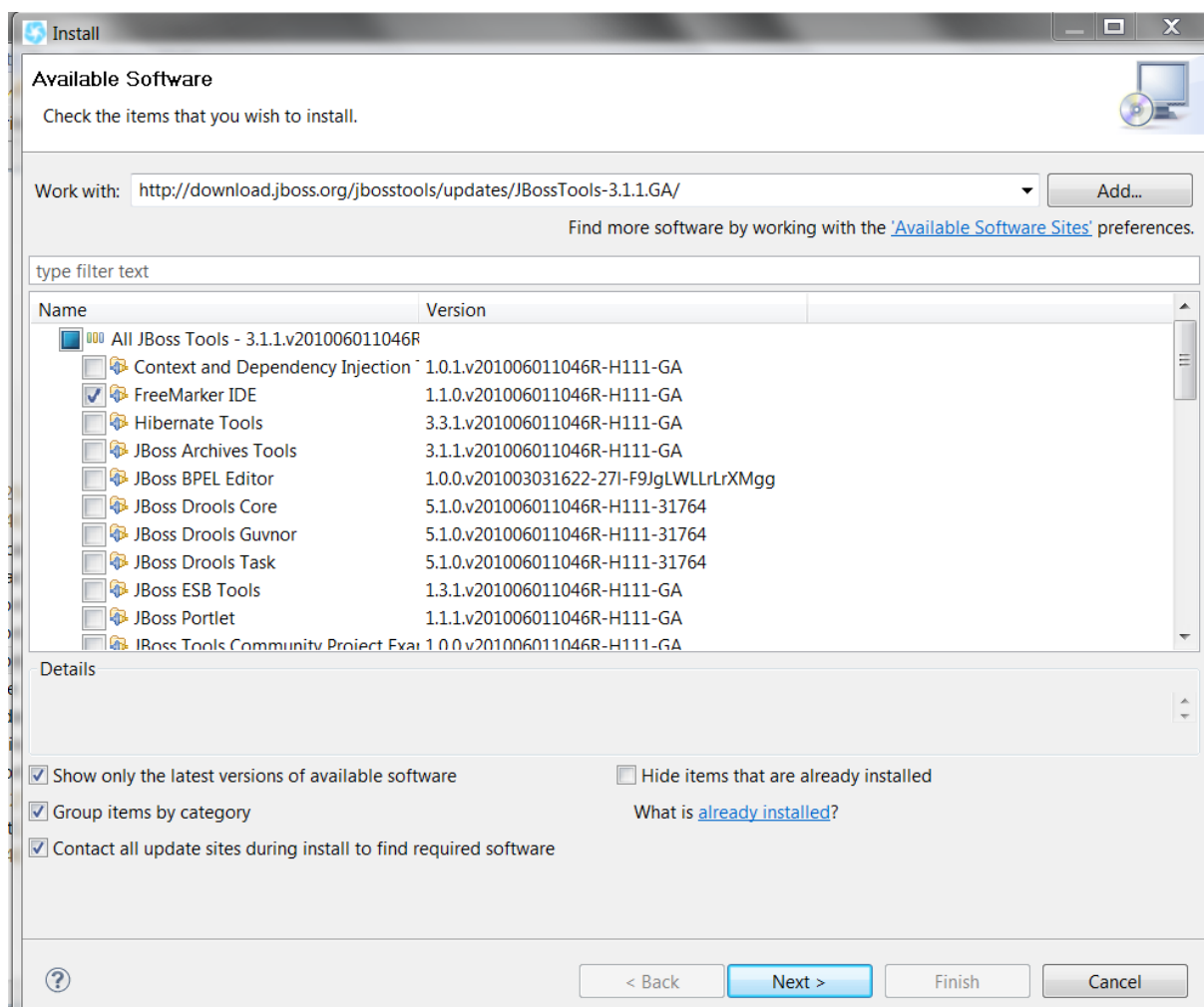


Illustration 19: Freemarker IDE Installation

- Accept license requirements and restart when needed

Configuration of maven projects

To create a maven projects, on the SIDE project name, right click and choose "SIDE->Create new Alfresco Module Project".

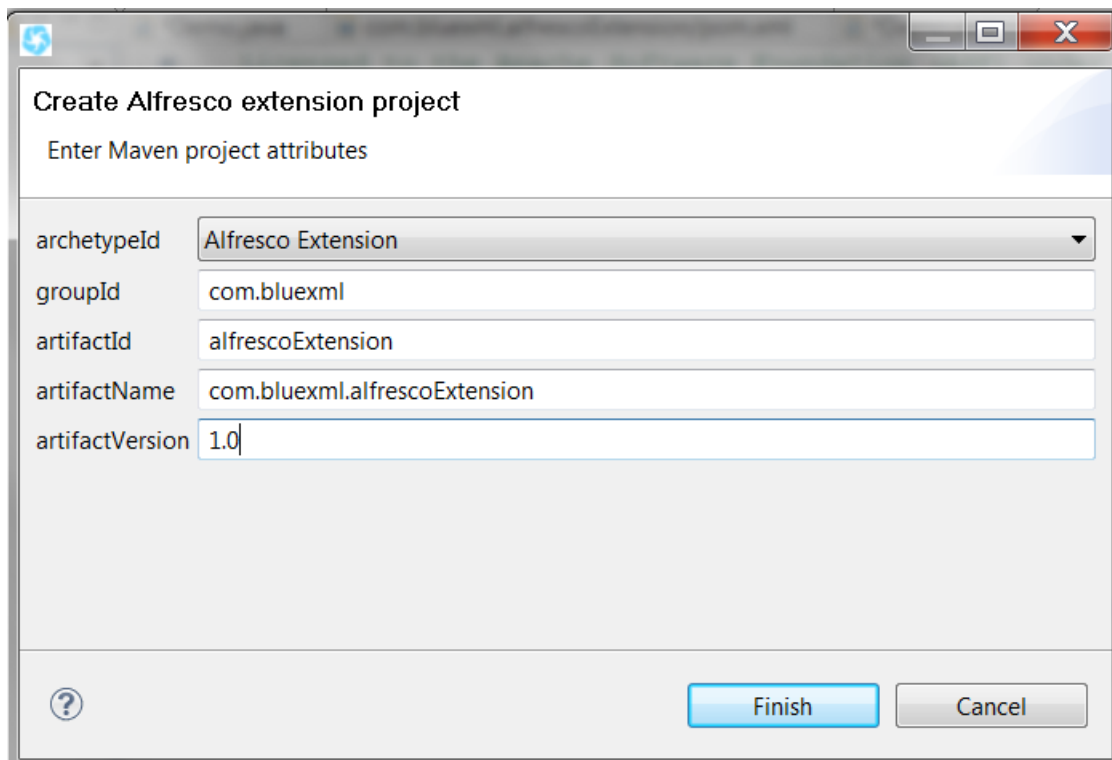


Illustration 20: Create Maven project module Window

In the current version of SIDE, two maven project archetype may be used:

- "Alfresco extension" to create a module which allows to extend the capabilities of the Alfresco webapp,
- "Alfresco Share extension" to create a module which allows to extend the capabilities of the Share webapp.

Select "Alfresco Extension" and enter the maven identification of your new module.

Click Finish to launch the creation of the maven module.

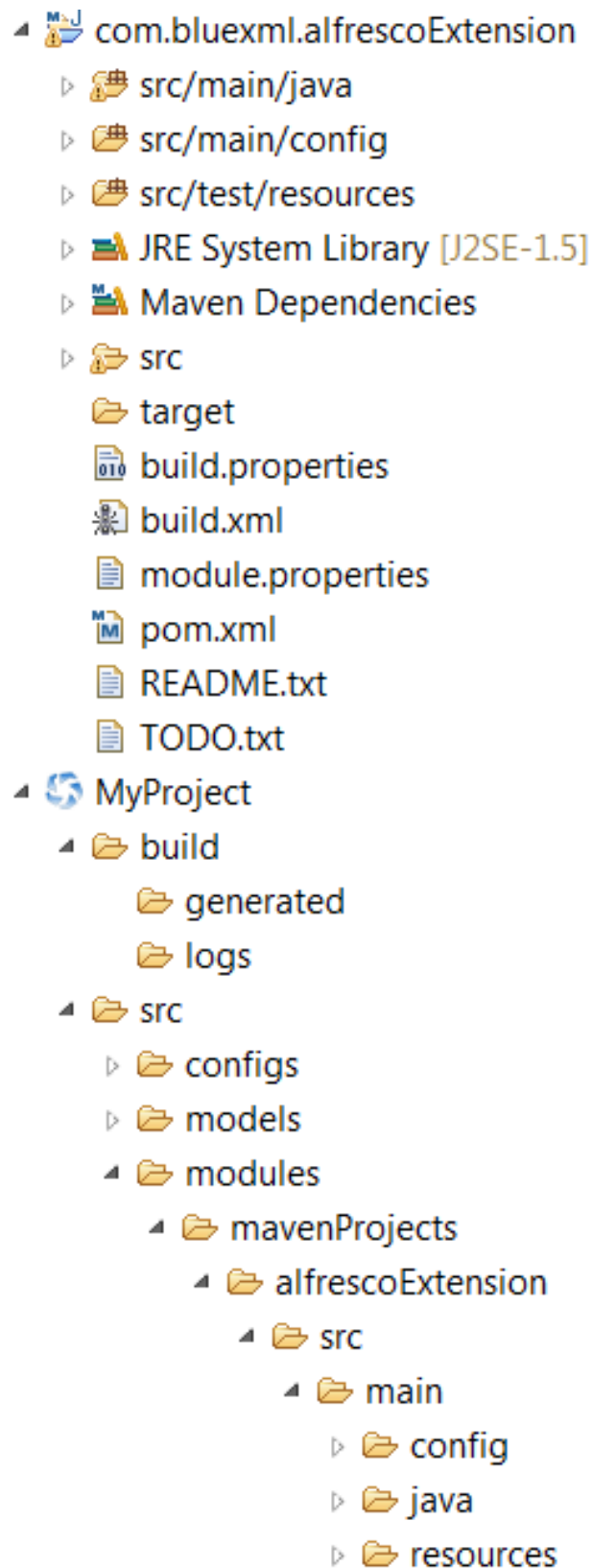


Illustration 21: New Maven module structure

Using the java perspective, you can now work on the new project.

If you want to add dependencies, open the `pom.xml` file using the Maven POM Editor:

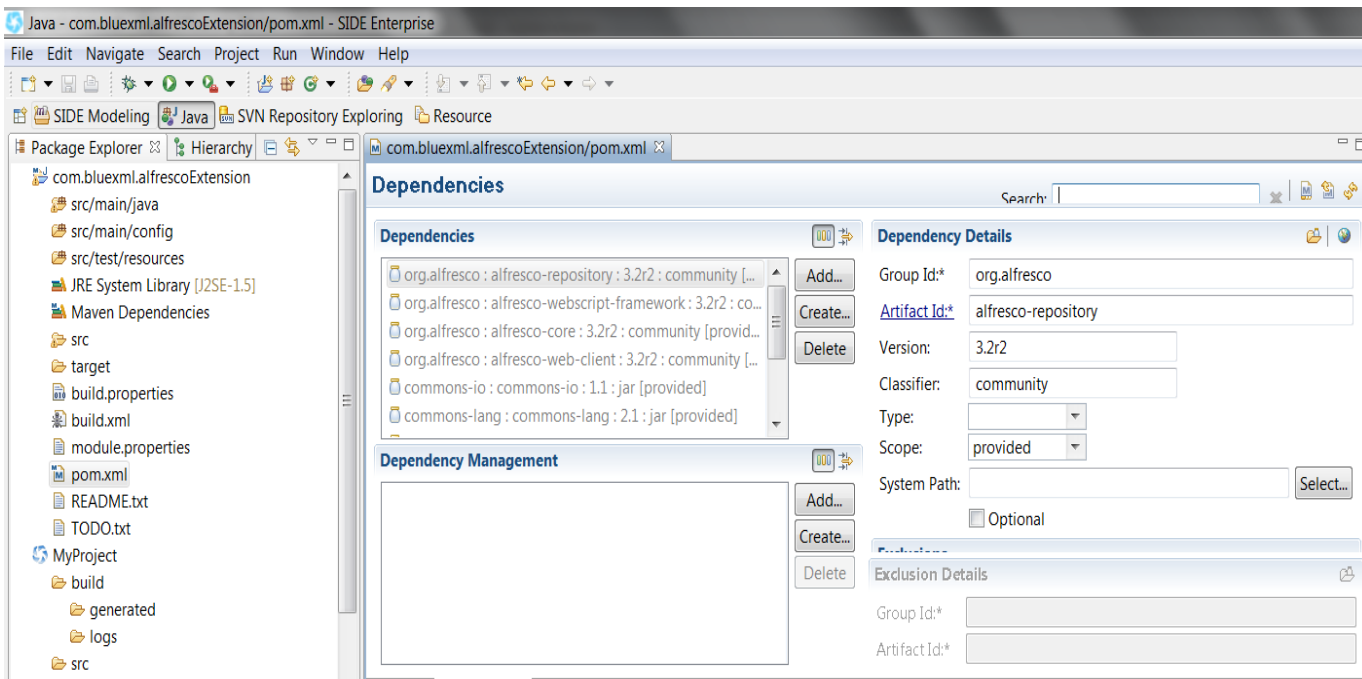


Illustration 22: POM Edition to add dependencies

According to the archetype, some dependencies have already been defined.

Note:

- In the current version of SIDE, if you want to add other dependencies, you need to manually install them in your local repository as a current bug prevent their autoamtiv installation.
- The current maven repositories used by SIDE does not provide the source package with the libraries. If you want to attach the source to the dependencies jar file, under the Maven Dependencies folder, right-click on the concerned jar and select 'Properties':

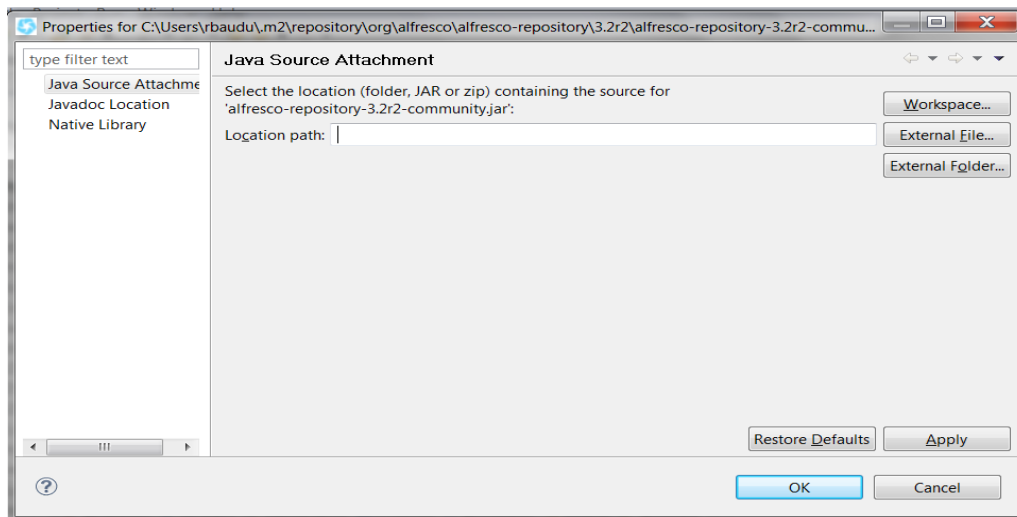


Illustration 23: Attach source to jar file

Give the path to the source folder.



SIDE FEEDBACK

In order to improve SIDE, you may participate to the feedback campaign: this campaign is absolutely anonymous and raises on the BlueXML web site various information on the way SIDE is used; this information is used by our product management team to make SIDE capabilities more accurate and more friendly.

If you want to help us improve SIDE:

- Click on Feedback
- Click on 'Change Upload Period Preferences'
- Click on Upload Always or other options more convenient to you.
- Check on 'I accept Term of Use'
- Click on OK

Change the Upload period to the value you think the most convenient to your way of using SIDE, particularly in term of frequency.

And many thanks for your help.



TARGETED FRAMEWORKS INSTALLATION

SIDE allows to generate and to deploy models on various frameworks. The following paragraphs give per frameworks useful installation and configuration information which will ease the integration with SIDE.

Alfresco

You can download the Alfresco release on the Alfresco website www.alfresco.com. You will get a file with a simple installation procedure.

Proceed with the installation.

In order SIDE may generate and deploy on this fresh installation of Alfresco, you need to keep in mind:

- for the SIDE generators related to the Alfresco technology, the Alfresco and Alfresco share webapp url. On a default installation,
 - Alfresco webapp url is `http://localhost:8080/alfresco`
 - Alfresco Share webapp url is `http://localhost:8080/share`
- for the SIDE deployer related to the Alfresco technology,
 - the Alfresco tomcat directory path. For instance, it may be on windows `"c:\alfresco\tomcat"`
 - the MMT Alfresco tool path for installing AMP packages. For instance, it may be on windows `"c:\alfresco\bin\alfresco-mmt.jar"`

Birt

You can download the Birt webapp application on the Birt website www.birt.org : you get a war file you can deploy in an application server container like the tomcat/webapp directory of Alfresco. The war will be automatically deployed as a web application by the application server.

Chiba

You can download the Chiba webapp application `xforms.war` on the side-labs download space http://sourceforge.net/projects/side-labs/files/_SIDE/SIDEforAlfresco/3.1.0a/SIDE-Run/: you get the `xforms.war` file you can deploy in an application server container like the tomcat/webapp directory of Alfresco. The war will be automatically deployed as a web application by the application server.

Note: actually, you can also get the Chiba webapp on the Chiba web site but the version we provide has been patched to avoid several issues which has not been reported in the Chiba 2.3 version and have only been fixed in the 3.0 release we do not support currently.

Facetmap

You can download the facetmap webapp application `facetmap.war` on the side-labs download space http://sourceforge.net/projects/side-labs/files/_SIDE/SIDEforAlfresco/3.1.0a/SIDE-Run/ : you get the `facetmap.war` file you can put in an application server container like the tomcat/webapp directory of Alfresco. The war will be automatically deployed as a web application by the application server.

ExtJS View and Forms Demo

You can download the ExtJS view and forms demo webapp application `side-demo.war` on the side-labs download space http://sourceforge.net/projects/side-labs/files/_SIDE/SIDEforAlfresco/3.1.0a/SIDE-Run/ : you get the `side-demo.war` file you can put in an



application server container like the tomcat/webapp directory of Alfresco. The war will be automatically deployed as a web application by the application server.

Workflow Demo

You can download the Workflow console demo webapp application wf_demo.war on the side-labs download space http://sourceforge.net/projects/side-labs/files/_SIDE/SIDEforAlfresco/3.1.0a/SIDE-Run/ : you get the wf_demo.war file you can put in an application server container like the tomcat/webapp directory of Alfresco. The war will be automatically deployed as a web application by the application server.



Table of Illustrations

<u>ILLUSTRATION 1: EXAMPLE OF MODELER: THE DATA MODELER.....</u>	<u>10</u>
<u>ILLUSTRATION 2: THE APPLICATION MODELER WINDOW.....</u>	<u>11</u>
<u>ILLUSTRATION 3: SIDE DEVELOPMENT AND COMPONENT SERVER ARCHITECTURE.....</u>	<u>12</u>
<u>ILLUSTRATION 4: JAVA ENVIRONMENT VARIABLES.....</u>	<u>14</u>
<u>ILLUSTRATION 5: SIDE MDA GENERATION COMMANDS.....</u>	<u>15</u>
<u>ILLUSTRATION 6: ERROR ON MISSING REQUIRED ARTIFACTS.....</u>	<u>16</u>
<u>ILLUSTRATION 7: SIDE MODELING PERSPECTIVE.....</u>	<u>18</u>
<u>ILLUSTRATION 8: SIDE PROJECT SELECTION.....</u>	<u>18</u>
<u>ILLUSTRATION 9: APPLICATION'S PROJECT CREATION.....</u>	<u>19</u>
<u>ILLUSTRATION 10: MODELS CREATION.....</u>	<u>19</u>
<u>ILLUSTRATION 11: APPLICATION'S PROJECT IN YOUR WORKSPACE.....</u>	<u>20</u>
<u>ILLUSTRATION 12: NEW MODELS WINDOW.....</u>	<u>20</u>
<u>ILLUSTRATION 13: SIDE DATA MODELER.....</u>	<u>21</u>
<u>ILLUSTRATION 14: APPLICATION MDA MANAGE CONFIGURATION WINDOW.....</u>	<u>22</u>
<u>ILLUSTRATION 15: GENERATION TAB OF THE MDA PROCESS.....</u>	<u>23</u>
<u>ILLUSTRATION 16: GALILEO UPDATE SITE.....</u>	<u>24</u>
<u>ILLUSTRATION 17: MAVEN AND DEVELOPMENT TOOLING TO SELECT</u>	<u>25</u>
<u>ILLUSTRATION 18: MAVEN PLUGIN INSTALLATION.....</u>	<u>25</u>
<u>ILLUSTRATION 19: FREEMARKET IDE INSTALLATION.....</u>	<u>26</u>
<u>ILLUSTRATION 20: CREATE MAVEN PROJECT MODULE WINDOW.....</u>	<u>27</u>
<u>ILLUSTRATION 21: NEW MAVEN MODULE STRUCTURE.....</u>	<u>28</u>
<u>ILLUSTRATION 22: POM EDITION TO ADD DEPEDENCIES.....</u>	<u>29</u>
<u>ILLUSTRATION 23: ATTACH SOURCE TO JAR FILE.....</u>	<u>29</u>

