

**ATL:**  
**Atlas Transformation Language**

**ATL Installation Guide**  
*- version 0.1 -*

**November 2005**

**by**  
*ATLAS group*  
*LINA & INRIA*  
*Nantes*

# Content

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
<b>2</b>	<b>Installing ADT from binaries .....</b>	<b>3</b>
2.1	<i>Installing Eclipse and EMF .....</i>	3
2.2	<i>Installing ADT.....</i>	3
2.3	<i>Additional required resources.....</i>	4
2.4	<i>First start of ADT.....</i>	4
<b>3</b>	<b>Installing ADT from source.....</b>	<b>5</b>
3.1	<i>Installing Eclipse and EMF .....</i>	5
3.2	<i>Installing ADT.....</i>	6
3.2.1	<i>Checking out ATL sources for Eclipse 3.0 .....</i>	7
3.2.2	<i>Checking out ATL sources for Eclipse 3.1 .....</i>	8
3.3	<i>Additional required resources.....</i>	9
3.4	<i>First start of ADT.....</i>	10
3.4.1	<i>Running ADT through the run-time workbench .....</i>	10
3.4.2	<i>Installing ADT in Eclipse from sources .....</i>	11
<b>4</b>	<b>Going further .....</b>	<b>12</b>
<b>5</b>	<b>References .....</b>	<b>12</b>

# Figures List

Figure 1.	Switching to the ATL perspective .....	4
Figure 2.	Creation of a new ATL project .....	5
Figure 3.	Configuration of the CVS access.....	6
Figure 4.	Checking ATL projects out.....	7
Figure 5.	Selection of projects to be refreshed.....	8
Figure 6.	Checking ATL debugging project for Eclipse 3.1 out.....	9
Figure 7.	Creating a new launch configuration .....	10
Figure 8.	Selecting a launch configuration.....	11
Figure 9.	Exporting ADT plug-ins .....	12



ATL Documentations

ATL Installation Guide

Date 01/12/2005

## 1 Introduction

The ATLAS Transformation Language (ATL) is the ATLAS research group answer to the OMG MOF [1]/QVT RFP [2]. It is a model transformation language specified both as a metamodel and as a textual concrete syntax. ATL is made available as part of the ATL Development Tools (ADT). The current ADT distribution runs over the Eclipse platform [3], and is based on the Eclipse Modeling Framework (EMF) [4].

This document aims to provide a step-by-step guide for installing the Eclipse/EMF distribution of the ATL Development Tools. Section 2 deals with the installation of ADT from binaries available from the GMT (Generative Model Transformer) project [5]. Section 3 details ADT installation from its source code (available on the Eclipse CVS repository). Finally, Section 4 provides some links to already available ATL resources, as well as references to the ATL mailing list.

## 2 Installing ADT from binaries

This section describes the ADT installation from binaries. Section 2.1 deals with the installation of Eclipse and the EMF plug-in. Section 2.2 describes the ADT installation from the binaries available from the GMT project web site. Section 2.3 details the installation of the additional resources that are required to run the ATL Development Tools. Finally, Section 2.4 describes the first start of ADT.

### 2.1 Installing Eclipse and EMF

The first step while installing ADT is to install both the Eclipse environment and the EMF framework. The last ADT release (version 20051102) is based on Eclipse 3.1.0 and EMF 2.1.0:

- Eclipse is available at <http://www.eclipse.org/downloads/index.php>. Select the release 3.1 ([eclipse-SDK-3.1-win32.zip](http://www.eclipse.org/downloads/index.php)).
- EMF can be downloaded from <http://download.eclipse.org/tools/emf/scripts/downloads.php>. Select the release 2.1.0 ([emf-sdo-runtime-2.1.0.zip](http://download.eclipse.org/tools/emf/scripts/downloads.php)).

**Important note:** users who want, for some reasons, to use a 3.0 Eclipse version must download the 2.0.2 version of the EMF plug-in.

Installing the Eclipse environment is simply achieved by unzipping the corresponding archive ([eclipse-SDK-3.1-win32.zip](http://www.eclipse.org/downloads/index.php)). This will create an \eclipse directory containing all the Eclipse files. At this stage, the EMF installation consists in unzipping the EMF archive ([emf-sdo-runtime-2.1.0.zip](http://download.eclipse.org/tools/emf/scripts/downloads.php)) into the same target directory. This last operation writes the set of EMF plug-ins in the \eclipse\plugins directory.

Next step consists in installing the set of ADT plug-ins.

### 2.2 Installing ADT

The ATL tools can be installed from binaries. This installation mode is recommended for those that only want to use ATL.

ADT binaries are available on the GMT (Generative Model Transformer) project web site. They can be downloaded from the "ATL Download Section". Current release build corresponds to the version 20051102 in the "Builds" subsection ([adt-20051102.zip](http://www.eclipse.org/downloads/index.php)).

**Important note:** users who want, for some reasons, to use a 3.0 Eclipse version must download the 20050916 version of the ADT archive.

Installing the ATL tools is achieved by unzipping the archive. **Be careful: the ADT archive has to be uncompressed into the \eclipse directory** that has been created by the Eclipse installation step (see Section 2.1), so that ADT plug-ins are placed into the \eclipse\plugins folder.

ADT plug-ins are now installed, but they still require some external libraries that are not included in the provided binaries. Next step of the ADT installation therefore consists in installing these additional resources that are required to run ATL programs.

### 2.3 Additional required resources

Beside the EMF framework, ATL is based on the ANTLR (ANother Tool for Language Recognition) [6] and the Netbeans MDR libraries (Metadata Repository) [7]:

- ANTLR can be downloaded from <http://www.antlr.org/download.html>. Select the “jar file only” (tool and Java runtime) archive ([antlr-2.7.5.jar](#)).
- MDR is available at <http://mdr.netbeans.org/download/>. Select the “MDR zip” archive ([mdr-standalone.zip](#)).

The **ANTLR library must be renamed into antlr.jar**. The library then has to be copied into the \eclipse\plugins\org.atl.eclipse.engine\_<version>\lib (for instance, \eclipse\plugins\org.atl.eclipse.engine\_1.0.7\lib).

The jar files contained in the MDR archive (jmi.jar, jmiutils.jar, mdrapi.jar, mof.jar, nbmdr.jar and openide-util.jar) have to be copied into the \eclipse\plugins\org.atl.engine.repositories.mdr4atl\_<version>\lib (for instance, \eclipse\plugins\org.atl.engine.repositories.mdr4atl\_1.0.0\lib).

Installation of the ATL tools is now completed. Section 2.4 provides some useful information for the first start of ADT.

### 2.4 First start of ADT

Now ADT have been installed into the Eclipse environment, it is time to start using ATL.

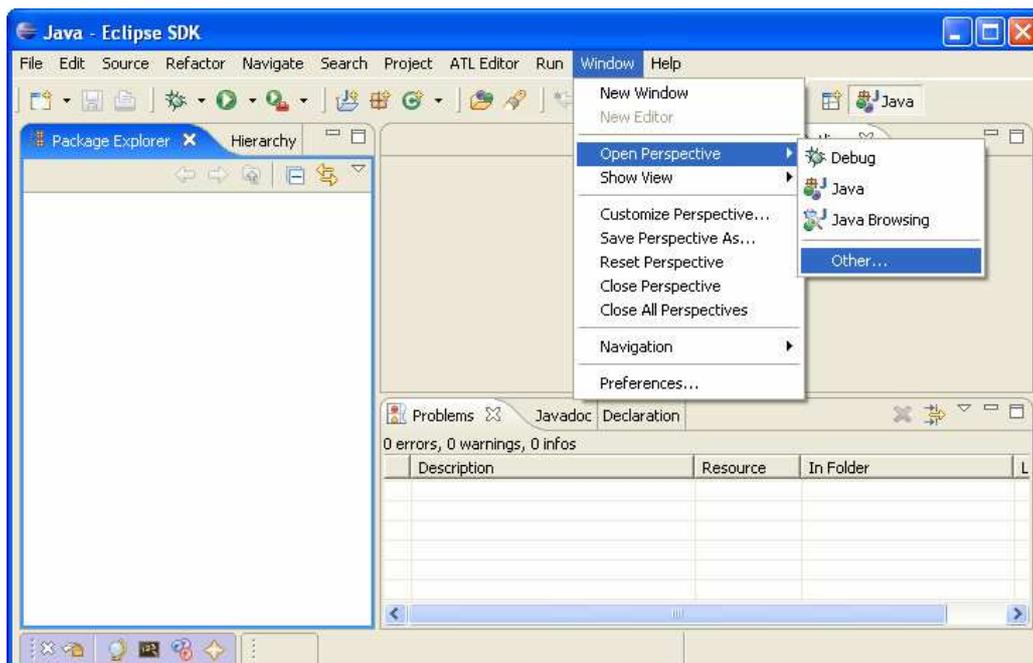


Figure 1. Switching to the ATL perspective

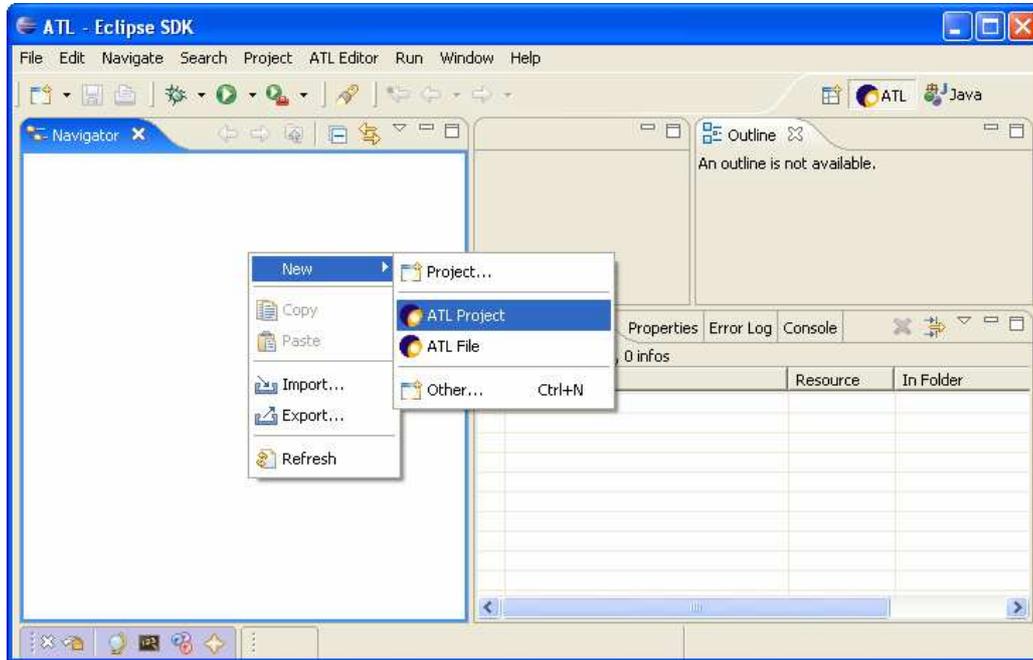


Figure 2. Creation of a new ATL project

Once Eclipse is launched, the first step is to switch from the current perspective (by default, the Java perspective is opened when Eclipse is started for the first time) to the ATL perspective. This could be achieved by selecting ATL in the perspective menu provided at Window→Open Perspective→Other... (see Figure 1). Be sure, when using ATL, to always be under the ATL perspective.

Once Eclipse has switched to the ATL perspective, it is possible to start using the ATL tools. The first step consists in creating a new empty ATL project in which ATL files (e.g. defining ATL transformations) can be defined. This creation can be achieved by mouse right click→New→ATL Project, as shown in Figure 2.

Section 4 provides a number of pointers to existing material on the ATL transformation language, including user manual, transformation examples...

### 3 Installing ADT from source

This section describes the ADT installation from its source code. Section 3.1 deals with the installation of Eclipse and the EMF plug-in. Section 3.2 describes the installation of the source code available on the Eclipse CVS repository. Section 3.3 details the installation of the additional resources that are required to run the ATL Development Tools. Finally, Section 3.4 describes the first start of ADT when installed from sources.

#### 3.1 Installing Eclipse and EMF

The first step while installing ADT is to install both the Eclipse environment and the EMF framework. The last ADT release (version 20051102) is based on Eclipse 3.1.0 and EMF 2.1.0:

- Eclipse is available at <http://www.eclipse.org/downloads/index.php>. Select the release 3.1 ([eclipse-SDK-3.1-win32.zip](http://www.eclipse.org/downloads/index.php)).
- EMF can be downloaded from <http://download.eclipse.org/tools/emf/scripts/downloads.php>. Select the release 2.1.0 ([emf-sdo-runtime-2.1.0.zip](http://download.eclipse.org/tools/emf/scripts/downloads.php)).

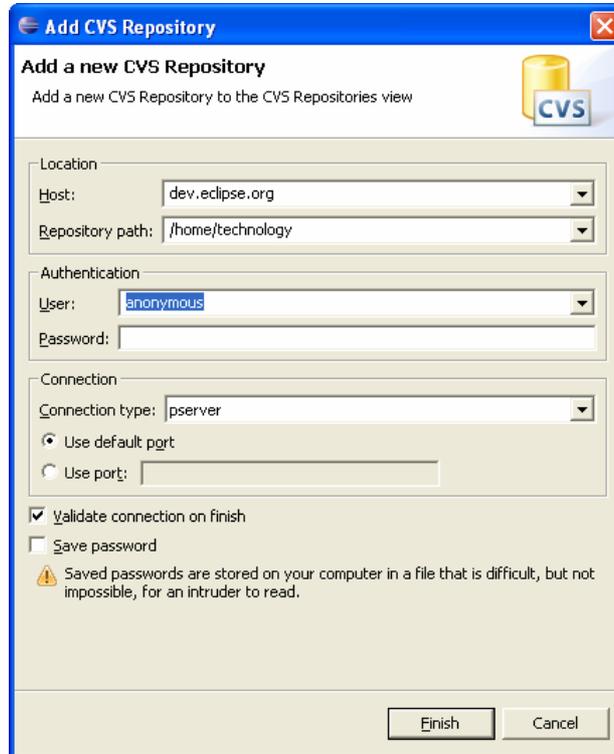


Figure 3. Configuration of the CVS access

**Important note:** users who want, for some reasons, to use a 3.0 Eclipse version must download the 2.0.2 version of the EMF plug-in.

Installing the Eclipse environment is simply achieved by unzipping the corresponding archive ([eclipse-SDK-3.1-win32.zip](#)). This will create an `\eclipse` directory containing all the Eclipse files. At this stage, the EMF installation consists in unzipping the EMF archive ([emf-sdo-runtime-2.1.0.zip](#)) into the same target directory. This last operation writes the set of EMF plug-ins in the `\eclipse\plugins` directory.

Next step consists in installing the ADT sources.

### 3.2 Installing ADT

The ATL tools can be installed from source code. This installation mode is recommended for developers that want to extend the set of tools.

The ADT sources are available onto the Eclipse CVS repository. It is assumed, at this stage of the installation process, that the Eclipse environment is installed (see Section 3.1). This document therefore details the ADT installation through the CVS repository view of the Eclipse environment.

The first step is to configure the CVS access. To this end, the user must move to the “CVS Repository Exploring” perspective. This could be achieved by selecting “CVS Repository Exploring” in `Windows→Open Perspective→Other....`

From this point, a new repository location has to be created: mouse right click→`New→Repository Location...` This opens the “Add CVS Repository” window (see Figure 3). The CVS repository must be configured as follows:

- the host is `dev.eclipse.org`;
- the repository path is `/home/technology`;

- the user is *anonymous*;
- no password is required;
- the connection type is *pserver*.

The repository location can then be created (click on “Finish”). It shall appear in the current “CVS Repository View” (as *pserver:anonymous@dev.eclipse.org:/home/technology*). Next step slightly differs according to the targeted Eclipse version.

### 3.2.1 Checking out ATL sources for Eclipse 3.0

From the created repository location, it is now possible to check the ADT projects sources out. To this end:

- open the HEAD→org.eclipse.gmt→ ATL directory;
- select the thirteen existing ATL projects:
  - org.atl.eclipse.adt.builder;
  - org.atl.eclipse.adt.debug;
  - org.atl.eclipse.adt.doc.developer;
  - org.atl.eclipse.adt.doc.user;
  - org.atl.eclipse.adt.editor;
  - org.atl.eclipse.adt.perspective;
  - org.atl.eclipse.adt.wizard;
  - org.atl.eclipse.engine;
  - org.atl.eclipse.km3;

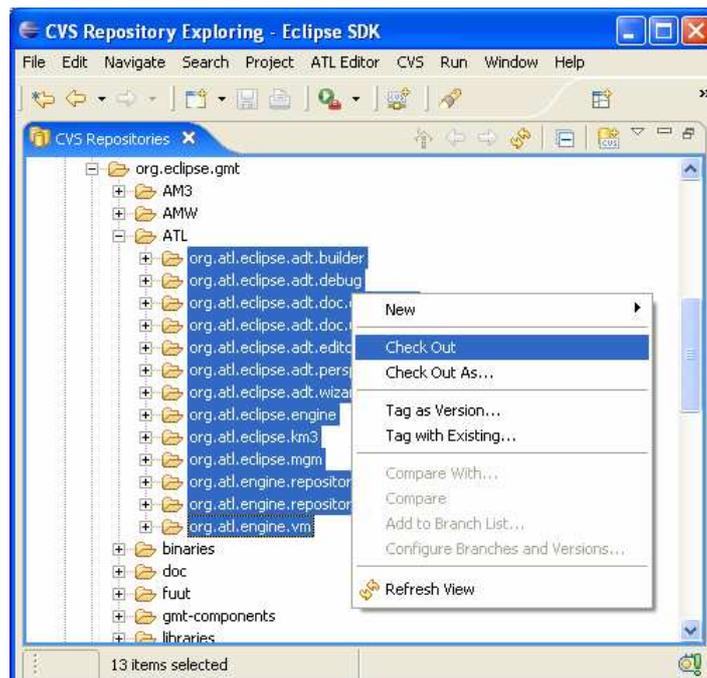
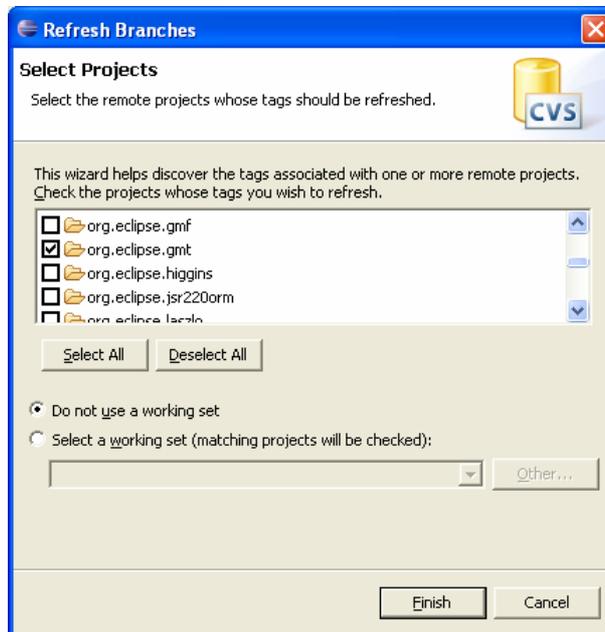


Figure 4. Checking ATL projects out

	ATL Documentations	
	ATL Installation Guide	Date 01/12/2005



**Figure 5. Selection of projects to be refreshed**

- org.atl.eclipse.mgm;
- org.atl.eclipse.vm;
- org.atl.eclipse.repositories.emf4atl;
- org.atl.eclipse.repositories.mdr4atl;
- check them out using right click→Check Out (see Figure 4).

This operation locally creates the corresponding projects. Created projects should appear when moving to the “Plug-in Development” perspective (Windows→Open Perspective→Other...→Plug-in Development).

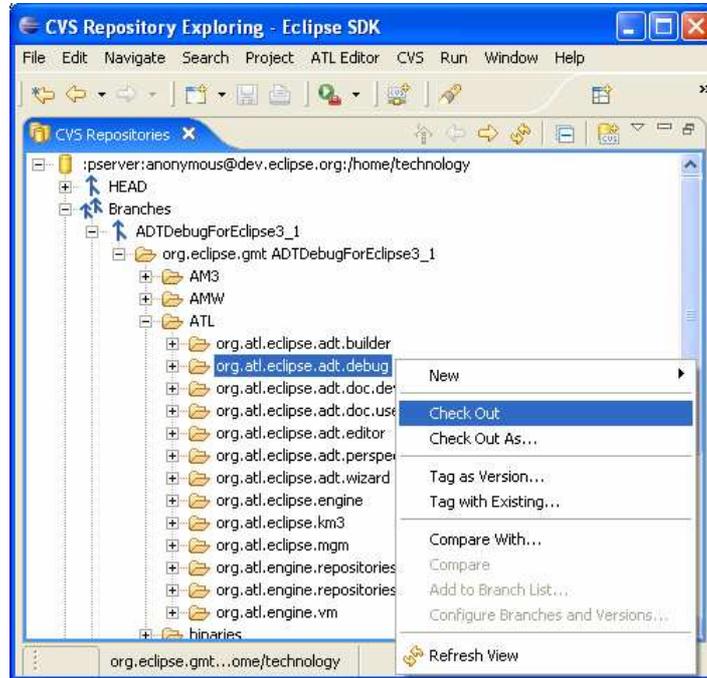
Note that, at this stage, errors appear in some of the checked projects (represented by a “red cross” in the “Package Explorer” view and detailed in the “Problems” view). These errors are due to the absence of some required external libraries that are not included in the downloaded plug-ins. Next step of the ADT installation therefore consists in installing these additional resources that are required to run ATL programs.

### 3.2.2 Checking out ATL sources for Eclipse 3.1

As for an Eclipse 3.0 platform, the ATL projects sources have to be checked out from the created repository location. However, there exists a debugging plug-in specific to Eclipse 3.1 versions that has to be checked out from a different branch of the repository location. Except this debugging project (org.atl.eclipse.adt.debug), the other twelve ATL projects must be checked out as described in Section 3.2.1.

Checking out the ATL debugging plug-in for Eclipse 3.1 is achieved as follows:

- if the “Branches” folder (of the repository location) appears to be empty, refresh it with mouse right click→Refresh Branches...:
  - this opens a “Refresh Branches” window (see Figure 5);
  - select the org.eclipse.gmt project and validate (click in “Finish”);



**Figure 6. Checking ATL debugging project for Eclipse 3.1 out**

- if a “Tags not found for some modules” window appears, select the “Search Deeply” option;
- the “Branches” folder should now contain two subfolders: “ADTDebugForEclipse3\_1” and “GMT0-1”;
- open the Branches → ADTDebugForEclipse3\_1 → org.eclipse.gmt ADTDebugForEclipse3\_1 → ATL directory;
- select the ATL debugging project:
  - org.atl.eclipse.adt.debug;
- check it out using right click → Check Out (see Figure 6).

Once this operation is completed, the corresponding projects should appear when moving to the “Plug-in Development” perspective (Windows → Open Perspective → Other... → Plug-in Development). It may be noted, in this perspective, that while the debugging project is connected to the “ADTDebugForEclipse3\_1 [dev.eclipse.org]” location, the other twelve ones are simply connected to “[dev.eclipse.org]”.

### 3.3 Additional required resources

Beside the EMF framework, ATL is based on the ANTLR (ANother Tool for Language Recognition) and the Netbeans MDR libraries (Metadata Repository):

- ANTLR can be downloaded from <http://www.antlr.org/download.html>. Select the “jar file only” (tool and Java runtime) archive ([antlr-2.7.5.jar](#)).
- MDR is available at <http://mdr.netbeans.org/download/>. Select the “MDR zip” archive ([mdr-standalone.zip](#)).

The **ANTLR library must be renamed into antlr.jar**. The library then has to be copied into the `\workspace_name\org.atl.eclipse.engine\lib` (for instance, `\workspace\org.atl.eclipse.engine\lib`).

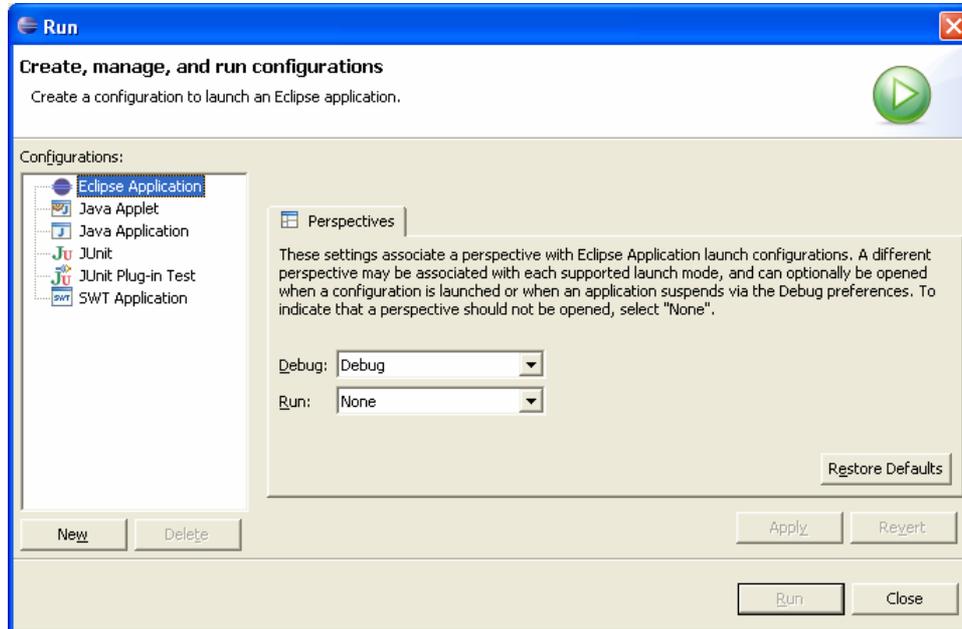


Figure 7. Creating a new launch configuration

The jar files contained in the MDR archive (jmi.jar, jmiutils.jar, mdrapi.jar, mof.jar, nbmdr.jar and openide-util.jar) have to be copied into the `\workspace_name\org.atl.engine.repositories.mdr4atl\lib` (for instance, `\workspace\org.atl.engine.repositories.mdr4atl\lib`).

Now these external libraries have been installed, the ATL projects can be built (right click→Refresh) without any error left. The installation of the ATL tools from their source code is now completed. Section 3.4 provides some useful information for the first start of ADT.

### 3.4 First start of ADT

Once the ADT installation from source has completed, there are two ways to run the ATL tools: running ADT through the run-time workbench or installing ADT into the Eclipse copy. These two options are detailed in the following subsections.

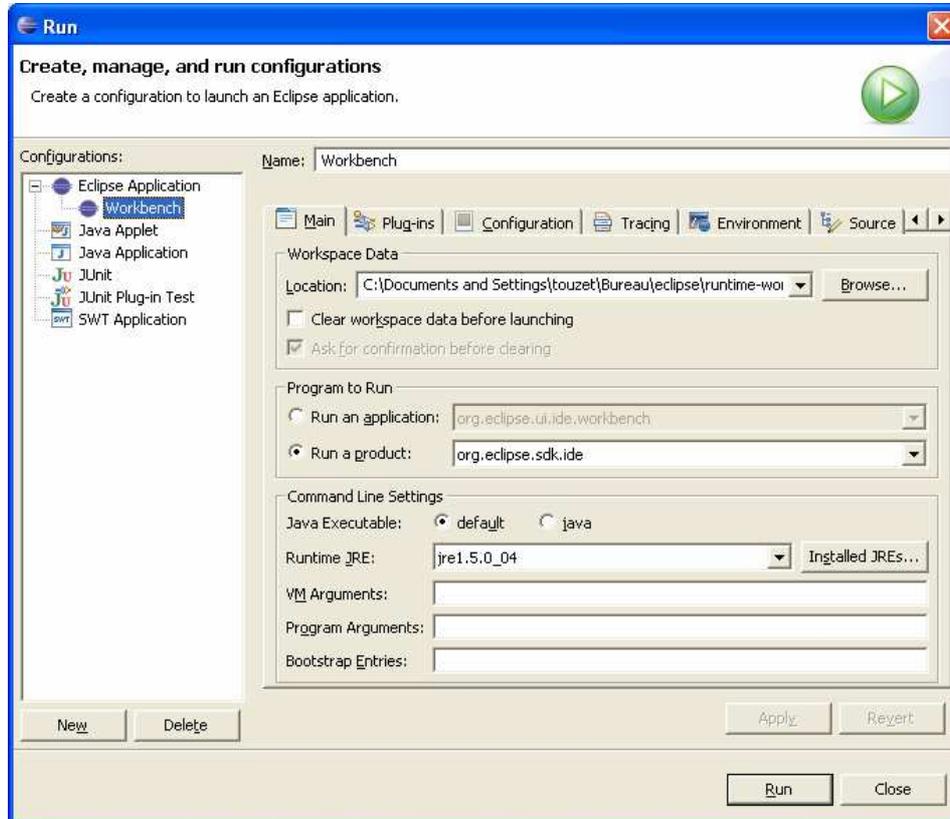
#### 3.4.1 Running ADT through the run-time workbench

The first option to run ATL from the downloaded source code is to execute the ATL plug-in projects with the run-time workbench. To this end, a new configuration enabling to launch the run-time workbench has to be created. This section focuses on the creation of a new run-mode launch configuration. However, developers wanting to extend the ADT functionalities should rather use a debug-mode launch configuration.

Creating a new run-mode launch configuration can be achieved by selecting the Run→Run... menu item. This selection opens a “Run” window (see Figure 7).

From this new window, select the “Eclipse Application” configuration and click on “New” to create a new launch configuration. Next step is to configure the launch configuration. The default configuration should be correct, but it may be interesting to rename the launch configuration to be created.

Once the new run-mode launch configuration has been created, the run-time workbench can be launched. Again, select the Run→Run... menu item, but now select the “Eclipse Application” launch configuration newly created (see Figure 8), and click on “Run” to start the run-time workbench.



**Figure 8. Selecting a launch configuration**

It is now possible to start working with the ATL transformation language. Please refer to Section 2.4 for information on the first steps with ADT.

### 3.4.2 Installing ADT in Eclipse from sources

The second way to run ATL from the downloaded source code is to install the ATL plug-ins into the Eclipse copy from the source code of the downloaded ATL projects. For this purpose, it is necessary to move to the “Plug-in Development” perspective (Windows→Open Perspective→Other...→Plug-in Development).

Once under the plug-in perspective, select the full set of ATL projects and choose the “Export” option in the contextual menu (right click→Export...). In the “Export” window, select “Deployable plug-ins and fragments”.

In the following screen (see Figure 9), make sure that all the ADT projects are selected, then select the archive file export destination, provide an archive name and start the export by clicking on “Finish”.

Next step consists in uncompressing the created ADT archive into the \eclipse directory. This operation generates the ADT plug-ins (in the \eclipse\plugins folder) corresponding to the downloaded ADT projects. As a last step, Eclipse has to be restarted.

It is now possible to start working with the ATL transformation language. Please refer to Section 2.4 for information on the first steps with ADT.

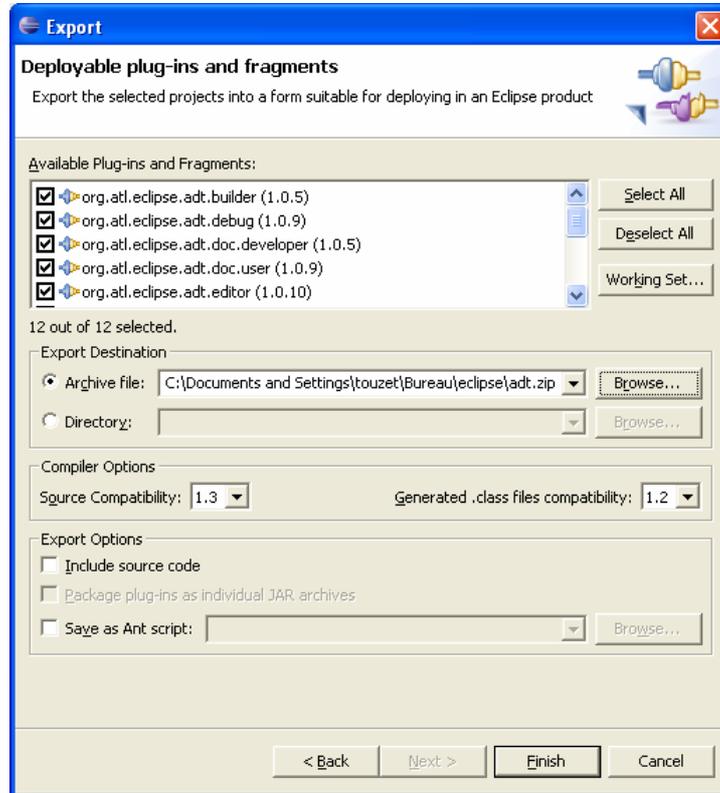


Figure 9. Exporting ADT plug-ins

## 4 Going further

The ATLAS Transformation Language (ATL) is part of the Generative Model Transformer (GMT) Eclipse project [5]. ATL users may find relevant resources on the GMT project site. These resources include the ATL FAQ, the ATL documentations (ATL User Manual, Specification of the ATL Virtual Machine) as well as many ATL transformations examples (in the ATL transformation part).

Moreover, there exists an ATL discussion board [8] enabling the ATL community to share information about ATL and the related model management tools.

## 5 References

- [1] OMG/MOF 2.0 Core Final Adopted Specification Document. ptc/03-10-04, 2004.
- [2] OMG MOF 2.0 Query / Views / Transformations RFP. ad/02-04-10, October 2002.
- [3] Eclipse home page. <http://www.eclipse.org/>.
- [4] Budinsky, F., Steinberg, D., Ellersick, R., Grose, T. Eclipse Modeling Framework. Addison Wesley Professional. ISBN: 0131425420, 2004.
- [5] The Generative Model Transformer (GMT) project. <http://eclipse.org/gmt/>.
- [6] The ANTLR project. <http://www.antlr.org/>.
- [7] The netbeans Metadata Repository (MDR) project. <http://mdr.netbeans.org/>.

	ATL Documentations	
	ATL Installation Guide	Date 01/12/2005

[8] The ATL mailing list. [http://groups.yahoo.com/group/atl\\_discussion/](http://groups.yahoo.com/group/atl_discussion/).