

# Guide to Building ArgoUML and ArgoUML-db

## Preface

This brief document describes how to build a complete ArgoUML-db package. However it's intended for users of the tool such as Software Engineering students and application developers rather than ArgoUML developers, who are thoroughly familiar with the process.

Users will find the initial effort involved in building the package proves well worthwhile. Model-driven design enables the software-class structure and database table relationships to be much more closely related to actual project requirements, and not lost in the low-level detail of SQL and Java (or other language) coding. The complete ArgoUML-db package includes a "design critic" which checks the graphical design for consistency, and SQL code and skeleton software-class code can be generated directly from the UML diagrams.

These notes have been written specifically for Linux, however most of it applies fairly directly to Microsoft Windows.

The prerequisite software for both Linux & Windows consists of the Java 6 SDK (not Java 5) and at least version 5 of the Apache Subversion (SVN) software version & revision control client. ArgoUML will require Java 6 in the near future and the -db plugin already needs Java 6, so Java 6 is assumed here. Apache SVN is open-source software and can be downloaded free of charge. Other tools required, such as ANT and the Xerces XML parser, are included in the "tools" directory of ArgoUML.

- 
1. It is necessary to build argouml first because building argouml-db requires some information from the argouml build.

When this document was written the most recent SVN release of argouml was 19545 and the most recent release of argouml-db was 80, and they built ArgoUML PRE-0.33.6. The current trunk releases of both packages should always be compatible.

Note, these notes may not completely apply to some earlier releases.

Information on an SVN revision can be obtained by revision number or date using SVN. This example uses the command-line client, but most clients also have a GUI:

```
# svn info [-r NUMBER] [-r "{YYYYMMDD}"] [etc]
http://argouml.tigris.org/svn/argouml/trunk --username
guest
```

2. Check out the current version of the ArgoUML source code from SVN into an empty directory with the following command, using a null password if prompted:

```
# svn checkout
http://argouml.tigris.org/svn/argouml/trunk --username
guest
```

For more detailed information refer to the Developer wiki <http://argouml.tigris.org/wiki> then go to "Building from source" under "Documentation" and Sc. 2.1 "Quick Start" for details.

3. Check out argouml-db from SVN into an empty directory, also with a null password:

```
# svn checkout
http://argouml-db.tigris.org/svn/argouml-db/trunk
--username guest
```

4. It's best to have a backup copy of all code before initiating any build as ArgoUML involves a very large download. So copy the downloaded argouml code into a build subdirectory, say <argoumlBuild/aBuild>, and the downloaded argouml-db code into another, say <argoumlBuild/dbBuild>.

5. Ensure the environment variable JAVA\_HOME points to the Java 6 JDK (though this may prove unnecessary, depending on the Java installation).

Then change to the “src” subdirectory and run <build.sh install> for a Linux installation or <build.bat install> for Microsoft Windows and the build will proceed. Any “warning” messages about deprecated usage may be ignored; only an “error” will result in an unsuccessful build.

The build process will finally display “BUILD SUCCESSFUL”.

If using the directory structure described in 4. above, the built package will be located in <path-to-argoumlBuild/aBuild/trunk/build> and it will contain 48 files and one “ext” subdirectory totalling about 15.8 Mbytes, though the exact details may change with releases.

6. Now we must build ArgoUML-db.

Set the PATH environment variable to include the “ant” tools package included with argouml and export it.. In Linux for example:

```
# PATH=$PATH:path-to-
argoumlBuild/aBuild/trunk/tools/apache-ant-1.7.0/bin
# export PATH
```

Change to the argouml-db directory containing <build.xml> and edit <module.properties> so “argouml.root.dir” points to the “argouml-build” subdirectory relative to there, for example:

```
argouml.root.dir=../../aBuild/trunk/src/argouml-build
```

The file <trunk/src/org/dbuml/base/model/DBModelFacade.java> may also be edited to remove cedilla accents in the word “Facade” in a few places, however these only appear in comments so the warning diagnostics can be ignored.

Then run:

```
# ant jar
```

7. This build process should also complete by displaying “BUILD SUCCESSFUL”.

Two “build” directories now exist: the argouml build in its directory tree and the argouml-db build at <path-to-argoumlBuild/dbBuild/trunk/build>

Copy the following files in the argouml-db build to the “ext” subdirectory of the argouml build at <path-to-argoumlBuild/aBuild/trunk/build> (later releases of argouml-db will do this automatically):

- o dbuml.jar from “trunk/build”
- o commons-collections-3.0.jar from “trunk/lib”
- o velocity-1.4.jar from “trunk/lib”

The file postgresql-8.2-505.jdbc2.jar should also be copied from <trunk/lib> if the PostgreSQL database is being used.

Copying these files to <ext> installs the plugin in the ArgoUML framework.

8. The ArgoUML “build” directory now contains a complete package which may be copied elsewhere and renamed, for example to “argoumlRun”. This “argoumlRun” directory may also be moved elsewhere as required.

---

### Running ArgoUML-db

The ArgoUML build for Linux includes a shell script <argouml.sh> which can be called via a soft-link in the /bin or other directory in order to start the application, though note it will need “execute” privilege if it's not set. This works even if the build directory is moved, however that requires a new soft-link of course.

ArgoUML can normally be started either by running “argouml.jar” or by calling “org.argouml.application.Main”. The former method is used in the shell script, however an initegration problem in the version used in this document requires that the “org.argouml.application.Main” form of startup be used if ArgoUML-db is included.

The startup script can be modified as follows, where the “java” command replaces the one in <argouml.sh>. If not familiar with the BASH command language, be sure to pay attention to the exact syntax, including the grave accents (or reverse single quote marks).

```
jdbc="path-to-the-MYSQL-JDBC-connector/mysql-connector-  
java-5.0.7-bin.jar"  
[...]  
argoMainPath="`dirname $PRG`/argouml.jar"  
argoDbPath="`dirname $PRG`/ext/dbuml.jar"  
java -cp $argoMainPath:$argoDbPath:$jdbc  
org.argouml.application.Main
```

This script can then be dragged and dropped as a logical link on the Desktop or anywhere else which is convenient.

**A.D.L.**

**9th June 2011**