

ArgoUML Quick Guide

Get started with ArgoUML 0.18

Kunle Odutola
Anthony Oguntimehin
Linus Tolke
Michiel van der Wulp

ArgoUML Quick Guide: Get started with ArgoUML 0.18

by Kunle Odutola, Anthony Oguntimehin, Linus Tolke, and Michiel van der Wulp

Copyright © 2001 Kunle Odutola

Copyright © 2004-2005 Michiel van der Wulp

This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Table of Contents

1. Introduction	1
1.1. Welcome to ArgoUML	1
1.2. About this guide...	1
2. Installing ArgoUML	2
2.1. System requirements	2
2.2. Installation options	2
2.3. Installing Auxiliary Modules	3
2.4. Command Line Options	3
2.5. Making .zargo-file clickable (on Windows)	3
3. ArgoUML Principles	5
3.1. Project, Model and Diagram	5
3.2. Objects	5
3.3. Overview of the window	5

Chapter 1. Introduction

Kunle Odutola

1.1. Welcome to ArgoUML

ArgoUML is a powerful yet easy-to-use interactive, graphical software design environment that supports the design, development and documentation of object-oriented software applications.

If you are familiar with a family of software applications called Computer Aided Software Engineering (CASE) tools then you should find ArgoUML instantly familiar.

The users of ArgoUML are software designers & architects, software developers, business analysts, systems analysts and other professionals involved in the analysis, design and development of software applications. Main features:

- Open standards: XMI, SVG and PGML
- 100% Platform independent thanks to the exclusive use of Java
- Open Source, which allows extending or customizing.
- Cognitive features like: reflection-in-action, opportunistic design, comprehension and problem solving

1.2. About this guide...

This guide - the ArgoUML Quick Guide - is provided to help you get up and running as quickly as possible with ArgoUML. It provides step-by-step instructions for obtaining, installing and starting ArgoUML, performing common tasks, and learn more about ArgoUML. If you need further details please look in the User Manual [<http://argouml.tigris.org/documentation/defaulthtml/manual/>].

Chapter 2. Installing ArgoUML

2.1. System requirements

Minimum system requirements:

- Any Operating System that supports Java.
- 10MB of free disk storage space on your hard disk.
- Mouse (or other pointing device) and a Keyboard.
- Java 2 JRE or JDK version 1.4 or higher.

2.2. Installation options

Now that you have a computer that can install and run ArgoUML effectively, you need to decide which of the installation options available is appropriate for you. You can choose from the following options:

- Java Web Start

This alternative is good for occasional users and tests. It is the easiest and quickest way to start using ArgoUML. It requires connection to the ArgoUML homepage [<http://www.argouml.org/>].

1. Make sure you have Java Web Start [<http://java.sun.com/products/javawebstart/>] installed.
2. Follow the Launch ArgoUML-link from the ArgoUML homepage [<http://www.argouml.org/>].
3. ArgoUML will be downloaded, cached locally and started.
4. On subsequent starts, the ArgoUML is available also from the Java Web Start console (even without being connected to the internet) and (if connected) at each start the version is checked against that on the web server and updates are automatically downloaded.

- Binary distribution

This is good for regular users and guarantees that the version of ArgoUML cannot change during the course of your project.

1. Make sure that you have a Java 2 JRE installed.
2. Download the ArgoUML binary distribution from the ArgoUML homepage [<http://www.argouml.org/>].

This could also include copying it to and from floppy disks or CD's in order to get it to the computer if no network connection is available.

3. Create an ArgoUML installation directory.
4. Change the current directory to the ArgoUML installation directory and extract the ArgoUML files to the installation directory.
5. Start by running the `argouml.jar` file, either by simply (double-)clicking on the file, or by

executing the following command: `java -jar argouml.jar` at the command line, or via a batch-file.

2.3. Installing Auxiliary Modules

The standard installation of ArgoUML does not support C++, php and C# code generation. To make this possible, you need to download the auxiliary modules from the download page at the ArgoUML website.

Unpack the downloaded file in the same directory as you unpacked ArgoUML itself.

The result should be that the directory that contains the `argouml.jar` file, now also contains a subdirectory named `ext`, which contains the `.jar` files for the extra languages.

2.4. Command Line Options

When starting ArgoUML from a command line, there are several extra possibilities. Try typing:

```
java -jar argouml.jar -help
```

You will see the instructions:

```
Usage: [options] [project-file]
Options include:
  -big           use big fonts
  -huge          use huge fonts
  -nosplash     don't display Argo/UML logo
  -noedem       don't report usage statistics
  -nopreload    don't preload common classes
  -profileload  report on load times
  -norecentfile don't reload last saved file
  -command <arg> command to perform on startup
  -batch        don't start GUI
```

You can also set java settings which influence the behaviour of ArgoUML:

```
-Duser.language [e.g. en]
-Duser.country  [e.g. US]
-Dforce.nativelaf [force ArgoUML to use the native look and feel. UNSUPPORTED]
```

A common problem is that the User Interface is shown in the wrong language. Below is an easy way to switch back to the English UI. Please be careful: the language is in lowercase, the country in uppercase!

```
java -Duser.language=en -Duser.country=US -jar argouml.jar
```

2.5. Making .zargo-file clickable (on Windows)

Ewan R. Grantham

This works only if you have installed using the binary distribution.

First, find a Zargo file, and right-click on it. You should get the normal Windows menu, including an Open or Open With option. At this point, give a description for the file like "ArgoUML Model", and tell Windows to use Notepad to open the file. That's not what you want to do ultimately, but it does get Win-

dows to accept the .zargo extension as representing a valid file type.

Now, open Windows Explorer and from the menu select View->Options (or on Windows XP: Tools->Folder Options...). You should get a two (or more) tabbed dialog. Click the "File Types" dialog, and then scroll through the list for the Description you gave - in the example "ArgoUML Model". Click to select that file type, and then click the "Edit" button.

Now, click on the "Open" action and then the "Edit" button there. You will then get a dialog that has a line for entering in the application action which will have a line for using Notepad to open the file. Replace that line with: "C:\Program Files\Java\j2re1.4.0_01\bin\javaw.exe" -jar "c:\ArgoUML\argouml.jar" "%L"

Substitute your path to your `javaw.exe` and `argouml.jar` files if they are in a different location. Click OK, then click OK, then click OK (three buttons on three different levels of the dialog).

The %L in the above commandline is needed instead of %1 because that is the way to support filename extensions of more than 3 characters.

Chapter 3. ArgoUML Principles

When ArgoUML starts, then it shows an empty class diagram on which you can add various objects. ArgoUML works according the following principles:

3.1. Project, Model and Diagram

The file operations save and open handle one `project` at a time. One `project` corresponds to a `model` plus diagram information, i.e. everything you can edit within the ArgoUML window.

The `model` may contain many objects (`ModelElements`) which form the complete UML description of the system you are describing. All `ModelElements` might be present on a diagram, but this is not required. Hence, the `model` that is stored in ArgoUML is independent of the contents of the diagrams. This may be explained by the possibility to generate programming code from the `model` - you do not need any diagrams for this.

A ArgoUML `project` also contains all diagram information, i.e. the shapes (presentation) used to represent the various UML `ModelElements`, their location, color, etc. Some `ModelElements` appear on multiple diagrams, some on one or none.

Hence, saving and opening projects retains all this information. There is a way to only save the `model` information though, which is by the menu "Tools" -> "Export as XML...". This may be usefull e.g. when generating programming code with an external tool that understands XML.

3.2. Objects

Select objects by left-clicking on them. The functionality of ArgoUML can be activated in the menu, in toolbars, or in pop-up menus when right-clicking above an object. Many of these functions work on the selected objects.

All diagrams have toolbars at the top which are used to create objects on the diagram.

Most objects can be added to a diagram and removed from a diagram without deleting it from the `model`! Select an object on a diagram, and then the menu item "Edit" -> "Remove from Diagram" clears the object from the diagram, but the object remains intact in the `model`, which may be seen in the explorer, i.e. the tree structure at the left hand side. Once removed, and item can be added back to the diagram (or any other diagram for that matter) by selecting it in the explorer and selecting the item "Add to Diagram" in the right-click pop-up menu.

3.3. Overview of the window

The top of the window contains a menu bar with commands available. In the `File` menu you can store the project or open another project instead.

The upper left part of the ArgoUML window shows a tree model of diagrams and objects. This view can be adapted to your needs by filtering the objects that are shown, and the structure in which they are shown.

The upper right part of ArgoUML shows the current diagram (one at a time). You can drag and drop the objects in the diagrams, and you can use the quick-links that appear when hovering over a selected object to create new objects connected to the already present objects.

The lower right part contains various details of the currently selected object: You select the object in one of the upper levels and choose what details you want to examine using the tabs.

The lower left part contains a list of all ToDo items for this model.