

ArtiSynth Quick Installation Guide

This describes how to quickly install one of the precompiled versions of ArtiSynth, which can be useful for users wanting to test the system and run the demonstration programs. More complete instructions, including how to obtain the latest development version from Github, compile and run using the Eclipse IDE, and install external model packages, are provided in the general installation guides for [Windows](#), [MacOS](#), and [Linux](#).


1 Instructions

1.1 To install ArtiSynth:

1. You need a 64 bit Windows, MacOS, or Linux system based on an Intel processor. You can also use systems based on the ARM processor if they implement an Intel compatibility layer. MacOS systems based on the new Apple ARM processor implement a compatibility layer (called Rosetta) that does appear to allow ArtiSynth to run as is.
2. Java should be installed. ArtiSynth will work with the most recent Java versions, but we recommend using the older Java 8 since that will also provide compatibility with MATLAB and the Jython console. You need a full Java Development Kit (JDK), not simply a runtime environment (JRE). We recommend the Java SE Development Kit 8uXXX (XXX being the latest update) which can be obtained from Oracle (registration required) at www.oracle.com/technetwork/java/javase/downloads/index.htm.
3. To verify that the Java JDK is visible to your system, open a terminal window (e.g., CMD on Windows), run the command `javac -version`, and check that the version matches the JDK. If it does not, follow the instructions in Section 2.
4. Download the latest precompiled ArtiSynth release (e.g., `artisynth_core_3.6.zip`) from www.artisynth.org/downloads.
5. Unzip the file into a folder, preferably one without spaces in the name.

1.2 To run ArtiSynth:

- On Windows: click `bin\artisynth.bat` under the install folder.
- On MacOS or Linux: use a terminal to run `bin/artisynth` under the install folder.

To run demos, open the Models menu at the top and choose the model you would like. It will be loaded and displayed in the viewer. To start/stop simulation, use the play button  at the top right.

1.3 To make changes to a demo:

1. If `<AT>` denotes the top-level folder of your ArtiSynth installation, then add `<AT>\bin` (Windows) or `<AT>/bin` (MacOS and Linux) to your system Path as described in Section 3.
2. Edit the `.java` file for the demo, which will be located in a folder under `<AT>\src\artisynth\demos`, using a plain text editor (e.g., notepad on Windows).
3. From within a terminal window (e.g., CMD on Windows), go to the demo folder and enter the command `compile`.

2 Making the JDK visible to your system

It is important to ensure that the JDK is visible to your system and supersedes any other Java installations.

Windows and Linux

If <JDK> denotes the top-level folder of your installed JDK, then add the folder <JDK>\bin (Windows) or <JDK>/bin (Linux) to your system Path, using the instructions given in Section 3. The folder should be added *ahead* of any other Java installations that might be specified on the Path.

On Windows, <JDK_DIR> is likely to be located under C:\Program Files\Java.

Mac OS X

On Mac OS X, you can set the “default” JDK by setting the JAVA_HOME environment variable. This can be done inside the initialization file for whichever command line shell you are using.

Assume that the desired JDK has version number 1.8.0_281 and that your home folder is <HOMEDIR>. Then for the bash shell, one can use a plain text editor to edit <HOMEDIR>/.bashrc and insert a line of the form

```
export JAVA_HOME='/usr/libexec/java_home -v 1.8.0_281'
```

while for the csh or tcsh shells, one can edit <HOMEDIR>/.cshrc and insert a line of the form

```
setenv JAVA_HOME '/usr/libexec/java_home -v 1.8.0_281'
```

Setting JAVA_HOME can also be done directly within the shell; doing it within the initialization file simply avoids the need to do so each time a new terminal window is opened.

3 Adding Directories to the System Path

The system “Path” is a list of directories which the system searches in order to find executables. Adding a directory to the path allows executables contained in that directory to be called directly from a command window (such as CMD on Windows).

Windows 10

1. Open the Start search, enter “env”, and choose “Edit the system environment variables”.
2. Click on Environment Variables.
3. Under User variables (the top window), click on Path and click Edit. If Path does not exist, click New.
4. In the Edit environment variable dialog, click New and enter the full path name for each directory you wish to add.
5. Close *all* dialogs by clicking OK and restart your command window.

Windows 8 and earlier

1. Right-click My Computer, and then click Properties.
2. Click the Advanced tab.
3. Click Environment variables.
4. In the top User variables window, click on Path and then Edit. If Path does not exist, click New.
5. In the edit window, add the full path name for each new directory, separated by semi-colons ‘;’.
6. Close *all* dialogs by clicking OK and restart your command window.

For example, if ArtiSynth is installed at C:\artisynth\artisynth_core and the desired JDK is at C:\Program Files\Java\jdk1.8.0_281, then we can add the bin directories for both by setting the User path to

```
C:\artisynth\artisynth_core\bin;C:\Program Files\Java\jdk1.8.0_281\bin
```

Mac OS X

Since Mac OS X is a Unix-based system, directories can be added to the path by editing the `PATH` environment variable directly in the initialization files for whichever command line shell you are using, as described in [Section 3](#).

On Mac OS X 10.8 and greater, directories can also be added to the path by adding a text file containing the directories to `/etc/paths.d`. In particular, we can create a file called `ArtiSynth` in `/etc/paths.d` that contains the full path names of the desired directories.

1. Open a terminal window
2. Use `sudo` to create `/etc/paths.d/ArtiSynth` with a plain text editor. For example:

```
sudo nano /etc/paths.d/ArtiSynth
```
3. Add the full path name of each desired directory, one per line, and save the file.
4. To test the revised `PATH`, open a new terminal window and enter the command: `echo $PATH`.

Linux

On Linux, directories can be added to the path by appending them to the `PATH` environment variable, which is a list of directories separated by colons `:`. The most direct way to do this is to redefine `PATH` inside one of the initialization files for whichever command line shell you are using.

Assume that your home folder is `<HOMEDIR>`. Then for the `bash` shell, one can edit `<HOMEDIR>/.bashrc` and insert a line of the form

```
export PATH=<DIR>:$PATH
```

while for the `csh` or `tcsh` shells, one can edit `<HOMEDIR>/.cshrc` and insert a line of the form

```
setenv PATH <DIR>:"$PATH
```