ANUGA Installation Guide

Release 1.3.7

February 17, 2015

Introduction

This document outlines the procedure for installing the Anuga toolbox. All components are licensed as open source and readily available from the net. It is assumed that the reader is familiar with the Python programming language and the process of downloading, installing and unpacking files into directories.

1.1   System requirements

To run ANUGA you will need a Windows PC or a Linux PC with at least 1GB RAM. As ANUGA is a memory-intensive numerical system, more memory is better than less.

The viewer (Windows only) requires a graphics adapter that is OpenGL compatible. It has been tested with ATI FireGL X1 cards and the NVIDIA family. It may not work with other cards such as those from the Intel(R) 82915G Express chipset family.

We recommend that you check out the page `http://anuga.anu.edu.au` for the latest information on installing Anuga.

1.2   Installation

Below are the install procedures for Windows 7 (32 bit) and Linux (32 and 64 bit).

1.2.1   Install - Windows (32 bit)

We use PYTHON as our programming environment together with a number of standard PYTHON packages such as numpy, scipy, matplotlib, netcdf4. One way to install all the required packages is to use a distribution like python_xy.

PYTHON xy

So first install python xy. This will be a large download, maybe 500 MB or more, but will provide a complete installation (well see NetCDF4 note below) of python for our needs . You should first remove any other version of python and mingw that may be on your system. The python xy package is currently only 32 bits, but this will still work on 64 bit Windows.

Be sure to choose the win32 python 2.7 version. This is the version for which we are developing. Netcdf4 Note

Check that netcdf is available. From a command line, try

```
python -c "import netCDF4"
```

If no error occurs then netCDF4 is available and you can disregard the rest of this note.

But unfortunately version 2.7.3.1 (April 2013) python xy seems to be missing netCDF4. Please let us know if later versions are ok. If it is missing then you need to install another package to cover this loss. For this we can use the precompiled scientific python binaries from `http://www.lfd.uci.edu/~gohlke/pythonlibs/#scientificpython`. I suggest you choose ScientificPython-2.9.2.win32-py2.7.exe to install. To test this install try:

```
python -c "import Scientific.IO.NetCDF"
```

64 bit

At the moment python xy is only 32 bit, but there seems to be a promise that a 64 bit distribution is not too far away. There is a 64 bit python distribution package from Enthought, but is free only to academic users. So at present we recommend win32 python xy. Manual install

It is possible to install the required environment manually. You would need to install Mingw to provide a compiler, the standard distribution of python27 and then precompiled python libraries for numpy, scipy, matplotlib, netcdf4 from a site like `http://www.lfd.uci.edu/~gohlke/pythonlibs/`. It would be interesting to hear feedback on this option, as there is an opportunity to use versions using the Intel Math Kernel Library, which should provide a useful increase in speed. Installing anuga

1.2.2   Obtaining ANUGA source

You can either use the source code avialble from sourceforge or download the latest version of the package using subversion.

You can obtain version 1.3 of the package from sourceforge by downloading the file `http://sourceforge/p/anuga`

1.2.3   Source from development repository

An alternative is checking out the most recent version of the package using subversion.

You need a subversion client. I suggest installing tortoise_svn downloads and then checking out the following svn repository. When you installed tortoise svn it creates a few extra menu items to your right click menu in the file manager. Just choose "tortoise" checkout to download the code.

```
https://anuga.anu.edu.au/svn/anuga/trunk/anuga_core
```

This should produce an anuga_core directory

1.2.4   Setup PYTHONPATH

We need to tell python where the anuga source code is located. This is done via the PYTHONPATH environment variable.

For instance, if your anuga_core directory was located at

```
C:\Users\Steve\anuga_core
```

then you should add

```
C:\Users\Steve\anuga_core\source
```

to your PYTHONPATH

Environment variables are accessed via control panel -¿ advanced system settings -¿ Environment Variables and then add a new Environment variable PYTHONPATH with value `C:\Users\Steve\anuga_core\source` (or what ever is appropriate for your installation)

1.2.5 Compiling ANUGA

Now go to the directory anuga_core and compile the anuga files. Fire up a cmd terminal, change to the anuga_core directory and run

```
python compile_all.py
```

Check that all the files have been compiled correctly. There should be an "OK" at the end of each separate compile command.

1.2.6 Run Unit tests

From the anuga_core directory run the unit tests via:

```
python test_all.py
```

1.2.7 Conclusion

Hopefully all the unit tests pass. As this is bleeding edge there are sometimes a small number of failures as this is a work in progress. Have a look at the demos in the directory anuga_core/documentation/user_manual/demos (along with the user manual) to see how to use anuga. Updating

1.2.8 Updating

If you downloaded the source using subversoin, you can update your version of ANUGA. This is fairly easy. Just choose the directory to "update" and then right click and choose "tortoise update" to update the code.

Then again from the anuga_core directory recompile the code and check the unit tests via

python compile_all.py python test_all.py

1.2.9 Quick install - Windows Vista and 7

The installation of the support software and the ANUGA software should be the same as for Windows XP above. The installation and use of ANUGA under Windows Vista and 7 have not been heavily tested. Feedback on any aspect of using ANUGA under Vista or 7 is welcomed.

1.2.10 Quick install - Linux

Please note that the following applies only to Ubuntu 10.04 (Ḷucid Lynẍ). We currently do not support any other Linux distributions.

Method 1

Choose the appropriate deb package for your architecture, download and use the package installer to install.

i386: `https://sourceforge.net/projects/anuga/files/anuga_ubuntu_package/python-anuga_1.2.0-0ubuntu3_i386.deb/download`
AMD64: `https://sourceforge.net/projects/anuga/files/anuga_ubuntu_package/python-anuga_1.2.0-0ubuntu3_amd64.deb/download`

Method 2

Open a terminal and enter:

```
sudo add-apt-repository ppa:anuga/ppa
sudo apt-get update
sudo apt-get install python-anuga
```

Method 3

Add the following lines to your `/etc/apt/sources.list`:

```
deb http://ppa.launchpad.net/anuga/ppa/ubuntu lucid main
deb-src http://ppa.launchpad.net/anuga/ppa/ubuntu lucid main
```

Download the ANUGA key from `https://datamining.anu.edu.au/anuga/raw-attachment/wiki/WikiStart/anuga.key`. You can add this key with:

```
sudo apt-key add anuga.key
```

or go to System -> Administration -> Software Sources -> Authentication -> Import Key File

Open a terminal and enter:

```
sudo apt-get update
sudo apt-get install python-anuga
```

1.3   Optional but recommended software

This software is not required to run the ANUGA toolbox, but it is recommended.

- psyco. Speeds up ANUGA by about 30%, so is strongly recommended. Under Ubuntu install either through Synaptic or by:

  ```
  sudo apt-get install python-psyco
  ```

  Under Windows install the file `psyco-1.6.win32-py25.exe` which you can find on the `http://psyco.sourceforge.net/download.html` page.

- VTK. The Visualization Toolkit. Under Ubuntu install either through Synaptic or by:

  ```
  sudo apt-get install python-vtk
  ```

Under Windows install the file `vtk-5.4.2-win32.exe` from the `http://www.vtk.org/VTK/resources/software.html` page.

1.4   Testing

In the ANUGA root directory, run the test suite:

```
python test_all.py
```

ANUGA has been succesfully installed if the tests pass as follows:

```
...
test_data_manager.py
test_interpolate_sww.py
test_mesh.py
test_mesh_interface.py
test_triangmodule.py
test_triangmoduleII.py
test_advection.py
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...........................
----------------------------------------------------------------
Ran 593 tests in 42.712s

OK
```

Note that if psycho is not installed, a number of psycho related warnings will appear. These warnings can be ignored.

Miscellaneous procedures

A.1 Setting the PATH on Windows

The Windows one-click installer should automatically set the necessary PATH environment "System variables". However, in case you do not have permission to modify the system variable, then the method of setting the PATH environment "User variable" for Windows XP is shown here. Setting the variable for Windows Vista should be similar.
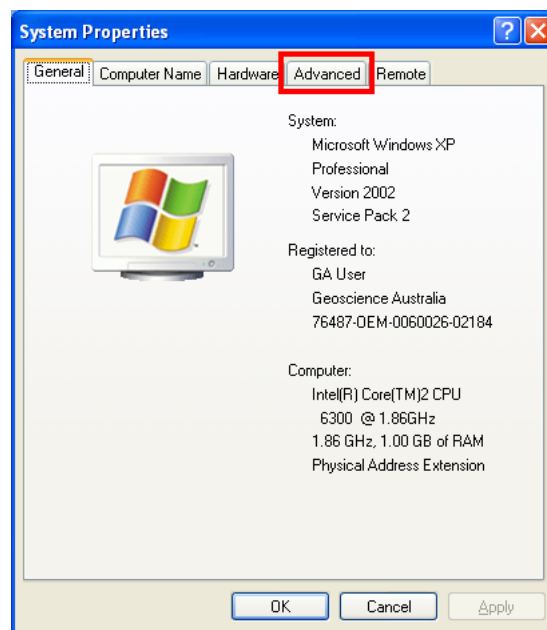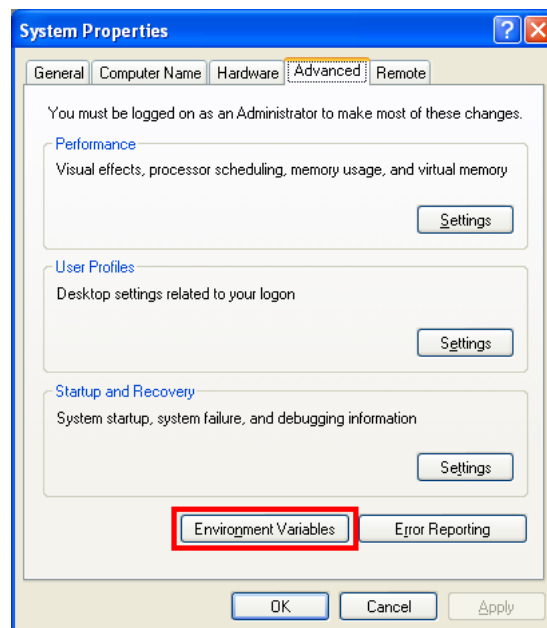
First, open the Control Panel:
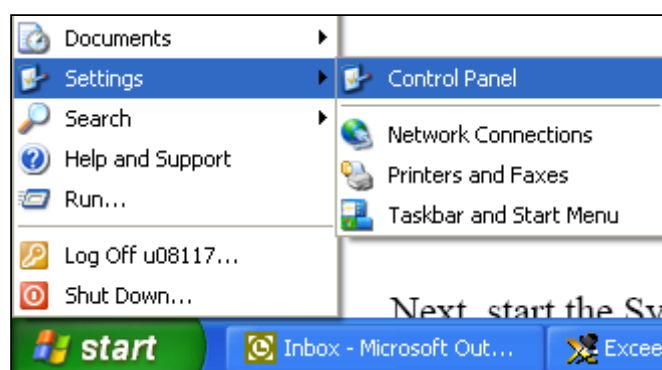
Next, start the System applet:
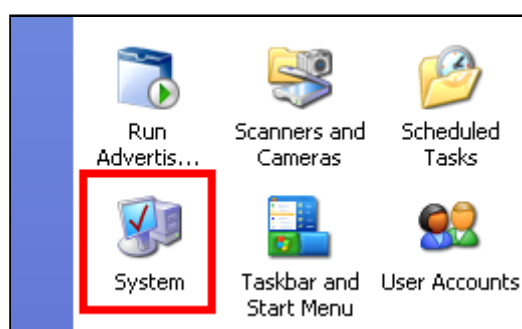
Select the `Advanced` tab in the System Properties window:

Press the `Environment Variables` button in the `Advanced` tab:

If the `PATH` variable is not defined in the 'User variables' or 'System variables' windows, press the `New` button in either of the two windows (for a personal machine, choose the 'System variables' window).

If `PATH` already exists in the 'User variable' or 'System variables' window, select the row with the `PATH` variable name in the appropriate window and press the `Edit` button next to the `New` button in that window:
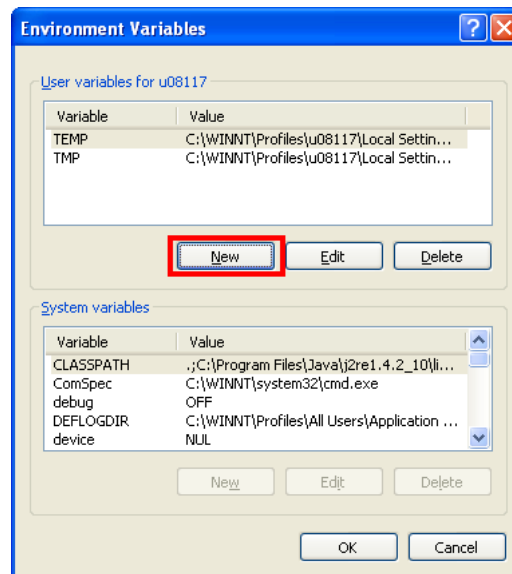
You will be shown the editor window whichever button you pressed in the above step. If the `Variable name` box is empty type in the name `PATH`. In the `Variable value` box type the value you want the `PATH` variable to have. If there is already some text in the box, place your additional value at the front of the existing value, not forgetting to terminate your additional string with the ';' character. The final value string must be a series of directory names seperated by ';' characters:
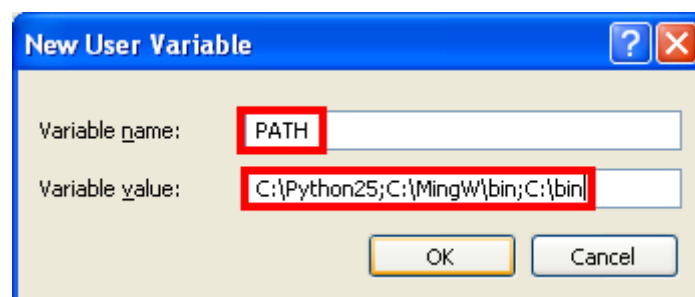

When you are finished, press the `OK` button and exit from the applet.

A.2 Associating animate.exe with a .sww file

The one-click installer should associate animate.exe with .sww files automatically. However, the method of associating `animate.exe` with a `.sww` file manually is shown here for Windows XP. A similar process should work for Windows Vista.
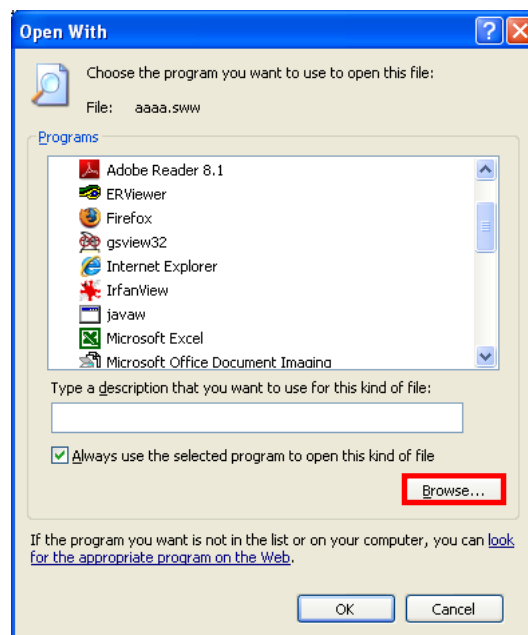
Double left-click on any `.sww` file. This brings up a dialog because Windows doesn't know how to open the file:
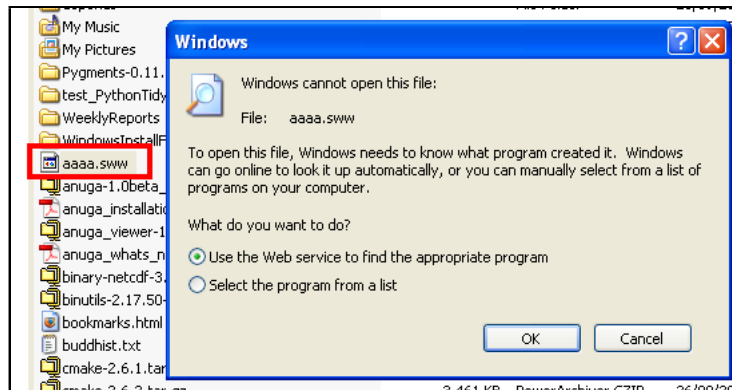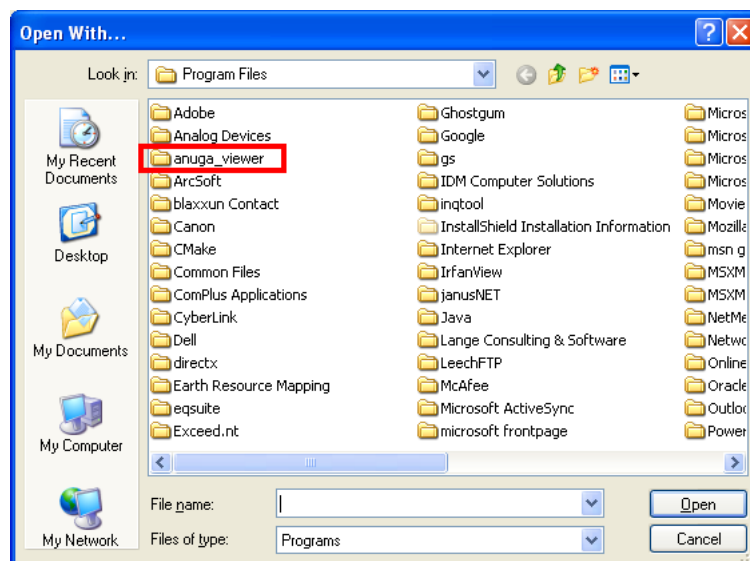

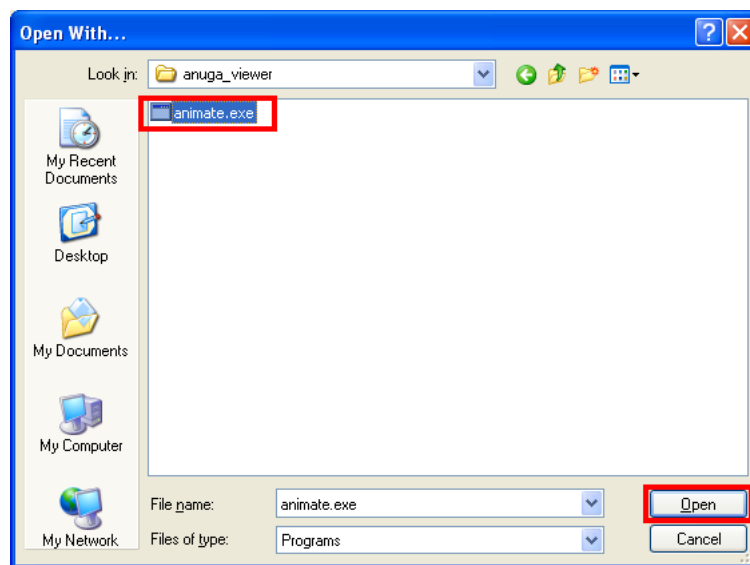Select the "`Select the program from a list`" radiobutton and press the `OK` button:
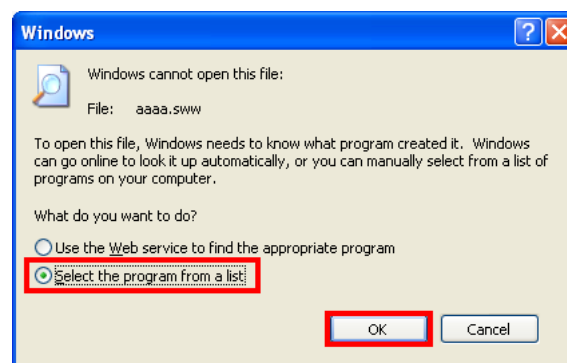
Press the `Browse...` button to find the `animate.exe` program:

Navigate to the `C:\Program Files\anuga_viewer` directory:

Select `animate.exe` and press the `Open` button:

Finally, press the `OK` button: