

---

# ANUGA Installation Guide

*Release 1.0beta\_4824*

Ole Nielsen, Duncan Gray, Jane Sexton

November 17, 2007

Geoscience Australia  
Email: [ole.nielsen@ga.gov.au](mailto:ole.nielsen@ga.gov.au)



---

# Introduction

This document outlines the required software 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 (2000 or XP) or a Linux PC with at least 512MB RAM. The viewer (Windows only) requires a graphics adapter that is OpenGL compatible. It has been tested with e.g. ATI FireGL X1 and the NVIDIA family. It may not work with cards such as those from the Intel(R) 82915G Express Chipset Family.

## 1.2 Quick install (Windows only)

This section provides the quickest path to getting ANUGA up and running on a Windows machine. For Linux and for more details please refer to Section 1.3.

1. Install Python pre-requisites: The Enthought version of Python2.4, Scientific Python and the NetCDF library<sup>1</sup>. These are available from either

- The supplied ANUGA distribution as under the folder `pre_requisites` (Install in the order `enthon-python2.4-1.0.0.exe`, `ScientificPython-2.4.9.win32-py2.4.exe` and the **unpack** `netcdf-3.6.1-beta1-win32dll.zip` into `C:\bin` as suggested below.
- The WEB:
  - Enthought version of Python2.4: <http://code.enthought.com/enthon/>.
  - Scientific Python:  
<http://sourcesup.cru.fr/frs/download.php/745/ScientificPython-2.4.9.win32-py2.4.exe>
  - NETCDF: download the file `netcdf-3.6.1-beta1-win32dll.zip` from <http://www.unidata.ucar.edu/software/netcdf/binaries.html> (<ftp://ftp.unidata.ucar.edu/pub/netcdf/contrib/win32/netcdf-3.6.1-beta1-win32dll.zip>). Unpack this in a folder that is available on the system path. We suggest unpacking it into

`C:\bin`

---

<sup>1</sup>Separate installation of Scientific Python and NetCDF will become unnecessary once Enthought sorts out netcdf support (see <https://svn.enthought.com/enthought/ticket/917>)

2. Add the following to the PATH environment variable

```
PATH=C:\Python24;C:\Python24\Enthought\MingW\bin;C:\bin
```

3. Unpack the ANUGA source code (anuga-1.0\_xxxx.tgz) in the python site-packages directory C:\Python24\Lib\site-packages. This is often done in Windows by double clicking on the tgz file and then select 'browse to' the destination directory.
4. In the ANUGA root directory (C:\Python24\Lib\site-packages\anuga), run the compilation script (either from the commandline or by clicking on it):

```
python compile_all.py
```

5. In the ANUGA root directory, run the test suite (e.g. by opening it with IDLE and running the file)

```
python test_all.py
```

ANUGA has been successfully installed if the tests pass.

6. To verify that ANUGA successfully reproduces a series of validation examples, go to the directory anuga\_validation and into automated\_validation\_tests in which you can run

```
python validate_all.py
```

To install the anuga\_viewer

1. Unpack the distribution (anuga\_viewer\_1.0.tgz) in

```
C:\Program Files
```

2. Double click on the test file

```
C:\Program Files\anuga_viewer\cylinders.sww
```

3. and associate .sww files with the executable

```
C:\Program Files\anuga_viewer\animate.exe
```

Try the demos provided in the ANUGA directory anuga\_demos (also discussed in the ANUGA user manual available with the distribution and also at [http://datamining.anu.edu.au/ole/anuga/user\\_manual/anuga\\_user\\_manual.pdf](http://datamining.anu.edu.au/ole/anuga/user_manual/anuga_user_manual.pdf)) and view the resulting sww files with the anuga\_viewer.

## 1.3 Pre-requisites (software)

This and the following sections is for custom installations and installations using Linux.

The ANUGA software core (shallow\_water, pmesh, least\_squares, ...) depends on the following software:

- python 2.3 or later <http://www.python.org/footnote>Note, the dll python2X.dll is needed to compile ANUGA, but is not always automatically added for versions 2.4 and above. Install it into the python2X directory, if you need it..
- python-numeric for Linux. It's called Numeric in Windows.
- python-scientific (to provide python-netcdf) for Linux. It's called ScientificPython in Windows.
- NETCDF. Extract the file netcdf.dll, and put in C://Windows//System32// (Windows XP) or C://WINNT//System32 (Windows NT or 2000).
- A C compiler such as gcc (from GNU in case of Linux and MinGW in case of Windows)

Note: All of these can be installed in one go on Windows platforms by using the Enthought Python distribution available at <http://code.enthought.com/enthon/>.

## 1.4 Recommended software

- psyco <http://psyco.sourceforge.net/index.html>: Speeds up anuga by about 30%.
- matplotlib (pylab) for quality 2d plotting (moving towards mandatory requirement and now part of Enthought.)
- VTK <http://mayavi.sourceforge.net/dwnld/vtk/win32/> - also now part of Enthought.

## 1.5 System configuration

The system path on Win32 architectures must contain the directories (this examples assumes the Enthought Python 2.4 distribution has been installed. Path names may vary depending on the exact configuration).

```
PATH=C:\Python24;C:\Python24\Enthought\MingW\bin;C:\Program Files\anuga_viewer
```

The path should be analogous for Unix operating systems.

If anuga is not installed in the python site-packages directory the root directory for all anuga/inundation modules must be on the pythonpath, e.g.

```
PYTHONPATH=C:\anuga_core\source
```

## 1.6 Installation of source zip file

Get a source zip file, for example, anuga\_source.3807.zip, and expand it in the python site-packages directory (C : Python24 Lib site-packages)

In the ANUGA root directory (C :  
Python24  
Lib  
site-packages  
anuga), run the compilation script:

```
python compile_all.py
```

## 1.6.1 Testing

In the ANUGA root directory, run the test suite:

```
python test_all.py
```

ANUGA has been successfully 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
```