

**Getting started:**  
**Installation and environment  
variables**

## Simple Installation 1

- Obtaining CDAT and untarring the sources
- The Climate Data Analysis Tools (CDAT) are developed primarily at the **Program for Climate Model Diagnosis and Intercomparison (PCMDI)** at the Lawrence Livermore National Laboratory (LLNL), but also by independent developers all the world.

## Simple Installation 2

- Because of the nature of the field in which CDAT is used, it will be ported to multiple operating system and hardware.
- For this reason the CDAT team decided to not distribute binaries since they're very likely to break even once ported to a very similar system.
- Moreover the CDAT is an open-source project. Because of these reasons the CDAT team decided to distribute only the sources, which then will need to be compiled on the users' machine(s).

## Simple Installation 3

- The latest sources can always be obtained from the CDAT website: <http://cdat.sf.net>. Nevertheless you might want to check the website to see if newer sources are now available. At the time this course is written, the distributed version of CDAT is 4.0.
- Unless you know what you're doing we recommend that you obtain the **cdat-everything** tarball.

## Simple Installation 3

- Once you've downloaded the sources you will need to untar the sources. On most Unix-like system this can be done with the following commands:

```
gunzip cdat-version-everything.tar.gz  
tar xvf cdat-version-everything.tar
```

- Once this is done you will need to change directory to *cdat-version* (`cd cdat-version`) and look at the `README.txt` file to see if your operating system requires you to set some environment variable before going any further.

## Compilers etc

Additional software not distributed with CDAT:

- Absolutely necessary is a C compiler, most Unix like system have one.
- Optionally some contributed Packages will require a FORTRAN compiler, while the GNU-FORTRAN 77 will do others can be used, users should check the Pyfort web-site to make sure their Fortran Compiler is supported. Known Linux Fortran compiler working with Pyfort/CDAT are: Intel Fortran Compiler, Portland Group Compiler, GNU\_Fortran.
- Also in order for the Visualization Control System (VCS) to output GIF files the ghostscript package is required.

## Express and Custom Installation

- There are basically 2 ways to install CDAT, the “express install” or the “custom install”.
- The express install will take care of everything for you and install on your system a new version of everything the CDAT needs.
- The custom install lets the user keep existing version of components and link CDAT with them. This is an expert mode and should be used very carefully.

# Express Install

- Overview

- This install options will install everything that CDAT needs on your system, including the latest versions of Python, Tcl/Tk, Numeric, Netcdf, OpenDAP, etc..., in a local directory specified by the person installing CDAT.

- Installation

- This is really easy, after making sure you set the necessary environment variables, simply do from the cdat-version directory (created when you untarred):

```
./express_install [your_path]
```

## System Tuning (Environment Variables)

- PYTHONPATH: finds module outside of python distribution. *Needs setting if multiple pythons or CDATs in your system (conflicts can arise!).*
- PYTHONSTARTUP: when type python, runs the file that this env variable points to. A good example of things to put in are the “Auto-completion” lines and the module and object you always need when starting a prompt session.

```
import rlcompleter
import readline
import cdtutil,genutil
readline.parse_and_bind("tab: complete")
```



Autocompletion with tab

```
import MA,cdms,vcs,MV
```



Modules you need often

```
x=vcs.init()
```



objects you need often

# PATH and PYTHONPATH

If you have other python (or CDAT) installations it is wise to:

- Set your environment variables based on:

```
PATH=$PATH:<your_cdat>/bin
```

```
# for executables
```

```
PYTHONPATH=$PYTHONPATH:<your_cdat>/lib
```

```
# for modules, packages and  
python extension modules
```

You can then run using 'cdat' or 'vcdat' or 'python'

# Examples

```
.profile* x .startup.py x
export PATH=./usr/local/cdat/bin:/usr/local/ba
bin:/usr/sbin:/sbin:/usr/bin/X11:/usr/kerberos/
grads/bin:~/wgrib_linux:/usr/local/badcdata::~~as
usr/local/ncu/bin

export PYTHONPATH=./usr/local/cdat/lib/python2
python:/usr/local/cdat/lib/python2.2/site-packa
python2.2:/usr/local/cdat/include/python2.2/Num
Numeric:~astephen/grib_reader_module:~astephen/
dynload:/usr/local/lib/badcdata:~astephen/bufr/
bufr/bufr_cdat/f2py/F2PY-2.32.225-1419/other:~a

export PYTHONSTARTUP=/home/tornado/astephen/.st
```

PYTHONPATH,  
PYTHONSTARTUP and  
PATH settings.

.startup.py script

```
.startup.py x
print "Importing Ag's startup script: .startup.py"
import os, string, re, sys

try:
    import cdms
    c=cdms.open
except:
    pass

if "DISPLAY" not in os.environ.keys():
    os.environ["DISPLAY"]="130.246.188.128:0"

try:
    import vcs
    v=vcs.init()
    p=v.plot
except:
    pass

pwd=os.getcwd()
ls=os.listdir(pwd)
cd=os.chdir

e40am="/badc/ecmwf-e40/.cdml/e40-1.0-am-1958-2001.xml"
e40ap="/badc/ecmwf-e40/.cdml/e40-1.0-ap-1958-2001.xml"
e40as="/badc/ecmwf-e40/.cdml/e40-1.0-as-1958-2001.xml"
e40at="/badc/ecmwf-e40/.cdml/e40-1.0-at-1958-2001.xml"
e40av="/badc/ecmwf-e40/.cdml/e40-1.0-av-1958-2001.xml"
e40fm="/badc/ecmwf-e40/.cdml/e40-1.0-fm-1958-2001.xml"
e40fs="/badc/ecmwf-e40/.cdml/e40-1.0-fs-1958-2001.xml"
```

## Tuning emacs for CDAT

- Emacs controlled with *.emacs*, you can get the latest python-mode for emacs on sourceforge website (or see <http://www.python.org>).
- Make sure your *.emacs* contains the following lines (assuming your python-mode file is in a directory named “emacs” under your home directory):

```
(setq load-path (cons "~/emacs/" load-path))
(autoload 'python-mode "python-mode" "Python editing mode." t)
(setq auto-mode-alist
      (cons ("\\.py$" . python-mode)
            auto-mode-alist))
```

```
(setq interpreter-mode-alist
      (cons ("python" . python-mode)
            interpreter-mode-alist))
```

- Other useful lines will be mentioned later.

**Appendix for session:**

**Getting started:  
Installation and environment  
variables**

# Custom (Expert) Install 1

- **Overview**

- In some case expert users will want to preserve components already existing on their systems, most of the time to use their existing Python where they already installed some custom Packages.
- It is possible (though not advised) to install CDAT separately from Python and the external Packages.

## Custom (Expert) Install 2

The CDAT distribution has been broken up in 3 distinct parts:

- **Python** and its external requirements (Tcl/Tk) – in the **pysrc/** sub-directory.
- the **external packages** required by CDAT (Numeric, Pmw, NetCDF, OpenDAP, R, VTK, pbmplus, gplot, etc...) – in the **exsrc/** sub-directory.
- and **CDAT Packages** themselves – in the “**Packages/**” (core CDAT) and “**contrib/**” (contributed packages) sub-directories.

## Custom (Expert) Install 3

- **Note**

- It is strongly recommended to not use this version since you might run into conflicting software version or have your software install in non-standard location which will make them difficult to find for **CDAT** or **contrib** Packages:

- **Installing Python only**

```
cd pysrc  
./install_script path/to/where/you/want/python
```

## Custom (Expert) Install 4

- **Installing External Software Required by CDAT**

If you wish to use your own Python and be your own version of some of the external software required by CDAT it is possible to do so.

- Installing **ALL** the external software:

- Let's assume your python is installed under **/usr/local** (i.e. the python binary is located at: **/usr/local/bin/python**)

```
cd exsrc
install_script /usr/local
```

- Installing **SOME** of the external software:

- Once again we're assuming Python is installed under **/usr/local**
- If you wish to install one package only (for example dods only)

```
cd exsrc
install_script /usr/local --dods-only
```

- If you wish to install **ALL but SOME** package (e.g. dods):

```
cd exsrc
install_script /usr/local --without-dods
```

## Custom (Expert) Install 5

- **Installing CDAT and the contributed packages:**

Here again you have multiple option installing all packages or just selected packages.

- Installing **ALL** packages:

Assuming your Python is in **/usr/local**

```
/usr/local/bin/python install.py
```

- To not install the contributed packages:

```
/usr/local/bin/python install.py -without-contrib
```

- Installing some CDAT packages

To install or reinstall an individual CDAT official Package, for example here the **vcs** package (once again we're assuming your python is **/usr/local/bin/python**):

```
cd Packages/vcs
```

```
/usr/local/bin/python setup.py install
```

To force rebuilding and install the package:

```
/usr/local/bin/python setup.py build --force install
```

- More info can be found in the documentation of distutils of Python.

# Installing Contributed Packages

- There are different ways to install or re-install an individual CDAT **contrib** Package, (once again we're assuming your python is in **/usr/local/bin/python**):
  - If the contributed package contains a **setup.py** file, follow the same procedure as an official CDAT Package.
  - If the contributed package contains an **install\_script** file, do:  
`./install_script /usr/local`
  - If the contributed package contains a **.pfp** file (**pyfort**) do:  
`/usr/local/bin/pyfort -i file.pfp`