PyRAF v1.1

Source Release

Installation Instructions

Science Software Branch
Space Telescope Science Institute

1.0 Quick Installation Directions

The quick instructions are for users, who have all required supporting packages present on their systems and their python installation works with all required modules enabled. In other words, this section is for those who know what they are doing and why they are doing it.

Unpack the source distribution file somewhere and build PyRAF and Numarray:

```
gunzip stsci_python_src_v1.1.tar.gz

tar -xvf stsci_python_src_v1.1.tar

cd stsci_python

python setup.py install

cd numarray-v0.6.2

python setupall.py install --gencode
```

This will install PyRAF, Pyfits, Numdisplay, Readgeis, Fitsdiff and Numarray in Python’s site-packages directory. Read Section 4.2 or type

```
python setup.py --help
```

to find how to do a personal installation.

2.0 Software Dependencies

While PyRAF is predominantly written in Python, it is dependent upon a number of external packages for support of its functionality. To use PyRAF, these packages will need to be resident on your system.

PyRAF depends upon the following support software packages.

1. IRAF
2. Tcl/Tk - v8.3 or higher
3. Readline (Optional, but highly recommended, needed for command line recall)
4. Python (with the readline and the Tkinter (_tkinter) modules enabled) - v2.2 or higher
5. Numeric - Python extension for manipulation of large arrays - v22 or higher

Tcl/Tk and Readline should be installed prior to installing Python, in order to be able to build the corresponding Python modules. Numeric is a Python extension and should be installed after a successful Python installation.

Note: This is the first source release which uses Python’s Distutils. We assume that you have all of the above packages already installed on your system.

The versions with which PyRAF has been most extensively tested can be downloaded from our web site:

http://www.stsci.edu/resources/software_hardware/pyraf/support_pyraf

Some platforms have most of these packages already installed in their system directories. To test whether your Python installation has all modules PyRAF needs, start Python and try to import them:

```
%python

>>>import readline

>>>import Tkinter

>>>Tkinter._test()

>>>import Numeric
```

If you get an `ImportError` for a module, then it is missing from your Python installation and you need to install it.

PyRAF on MacOSX was tested with the supporting packages installed both through fink v0.5.3 and from sources on the Unix command line. We find that generally installations using fink are easier. All supporting packages can be installed through fink, including Numeric.

You will also need a C compiler and the X11 windowing system. On MacOSX compilers can be installed with the Developer's tools, available from:

http://developer.apple.com/tools/download/

X11 is required by IRAF. Please, follow the directions on their site to get IRAF working on your system.

If all of the above tests pass and you have all necessary components installed, then you can skip the next section and read about installing all packages in stsci_python in Section 4. In case any of the modules cannot be imported, please read on. Next section gives concise directions on how to install each of the supporting packages, however if you run into serious problems, you will need to consult the detailed installation instructions for that particular package.

3.0 Installing The Supporting Packages

IRAF installation is not discussed in this document. IRAF is maintained by the iraf group at NOAO. For installation directions or problems, go to their web site **http://iraf.noao.edu** or contact them at **iraf@noao.edu**.

MacOSX installations: IRAF on MacOSX can be installed using the directions on the IRAF web site or using a package installer from this site:

http://www.owlnet.rice.edu/~marcosh/iraf/

We recommend that MacOSX users use fink to install all other packages. Our experience shows that installation of all supporting packages through fink v0.5.3 is generally very easy. Fink can be downloaded from:

http://fink.sourceforge.net/

Some of the packages don't have binaries available from fink. We found it is easiest to use `FinkCommander` for all installations, rather than using `dselect` or `apt-get`. This will do a source installation, the only drawback is that it takes a little longer for the installation to complete.

Source Installations: If installation from source is necessary, on most systems the following will work:

- To unpack a source file
  %gunzip package.tar.gz
  %tar -xvf package.tar
- To configure and build a package:
  %cd package
  %./configure --prefix=<install-dir>
  %make
  %make install

This will create directories `bin, lib, include` under `<install-dir>`.

If you install any of these packages in a personal directory, most likely, you will need to change (or set) two environment variables:

```
setenv  LD_LIBRARY_PATH <install-dir>/lib

setenv PATH <install-dir>/bin
```

3.1 Tcl/Tk

It is very likely that Tcl and Tk are already installed on your system. Look in the system directories for files like libtcl.* and libtk.*. Python requires Tcl/Tk v8.3 or later.

On a Linux system, the rpms for these packages are on the installation CD. You will need the libraries and the header files installed.

If you have to build these packages from source, build them as shared libraries. On most systems the following set of commands will work for Tcl and Tk:

```
%cd tcl8.3.5/unix

%./configure --enable-shared --prefix=<install-dir>

%make

%make install
```

3.2 Readline

Readline is probably already installed on your system. In case you need to install it in your personal directories, the following commands will install Readline on most systems:

```
%cd readline

%./configure --prefix=<install-dir>

%make

%make install
```

3.3 Python

Linux: More recent versions of Redhat come with Python 2.2 installed. Tkinter is installed separately from the installation CD. If you prefer to use a newer version of Python, you can try to find RPMS for it. Note that you will also need RPMS for Tkinter (for that version of python and your version of the operating system).

Source Installation: If you need to install from source, download it from our web page and unpack it. If Tcl/Tk and Readline libraries are on LD_LIBRARY_PATH or in a system directory, the next three commands are usually sufficient to install Python from source:

```
%cd python
```

```
%./configure --prefix=<install-dir>
```

Only for **Redhat 9.0,** an additional option is needed:

```
%./configure --enable-unicode=ucs4 --prefix=<install-dir>
```

```
%make
```

```
%make install
```

To test whether your Python installation has all modules required for PyRAF enabled, try to import the modules as described in Section 2.0.

In some cases, if Tkinter does not build by default, you may need to edit the file Modules/Setup in the Python source distribution, to let Python know where Tcl/Tk and X11 libraries are. Below is an example of this section of the Setup file on Solaris. Note, that some lines are uncommented and the paths on your system may be different.

```
# The _tkinter module.
#
# The command for _tkinter is long and site specific.  Please
# uncomment and/or edit those parts as indicated.  If you don't have a
# specific extension (e.g. Tix or BLT), leave the corresponding line
# commented out.  (Leave the trailing backslashes in!  If you
# experience strange errors, you may want to join all uncommented
# lines and remove the backslashes -- the backslash interpretation is
# done by the shell's "read" command and it may not be implemented on
# every system.

# *** Always uncomment this (leave the leading underscore in!):
 _tkinter _tkinter.c tkappinit.c -DWITH_APPINIT \
# *** Uncomment and edit to reflect where your Tcl/Tk libraries are:
-L/usr/local/lib \
# *** Uncomment and edit to reflect where your Tcl/Tk headers are:
-I/usr/local/include \
# *** Uncomment and edit to reflect where your X11 header files are:
#-I/usr/X11R6/include \
# *** Or uncomment this for Solaris:
-I/usr/openwin/include \
# *** Uncomment and edit for Tix extension only:
#-DWITH_TIX -ltix8.1.8.2 \
# *** Uncomment and edit for BLT extension only:
#-DWITH_BLT -I/usr/local/blt/blt8.0-unoff/include -lBLT8.0 \
# *** Uncomment and edit for PIL (TkImaging) extension only:
#     (See http://www.pythonware.com/products/pil/ for more info)
#-DWITH_PIL -I../Extensions/Imaging/libImaging  tkImaging.c \
# *** Uncomment and edit for TOGL extension only:
#-DWITH_TOGL togl.c \
# *** Uncomment and edit to reflect your Tcl/Tk versions:
-ltk8.3 -ltcl8.3 \
# *** Uncomment and edit to reflect where your X11 libraries are:
#-L/usr/X11R6/lib \
# *** Or uncomment this for Solaris:
-L/usr/openwin/lib \
# *** Uncomment these for TOGL extension only:
#-lGL -lGLU -lXext -lXmu \
# *** Uncomment for AIX:
#-lld \
# *** Always uncomment this; X11 libraries to link with:
-lX11
```

3.4 Numeric

Once Python is installed and configured, installing Numeric is straightforward. Unpack the tar file, go to the `Numeric` directory and install it:

```
%python setup.py install
```

This will install Numeric in Python’s `site-packages.`

4.0 Installing All Packages in stsci_python

To fully support all tasks in STSDAS, this source distribution is packaged with source releases of PyRAF v1.1, Numarray v0.6.2, Pyfits v0.8, Readgeis v1.6 and Numdisplay v0.1beta2.

Download the file **stsci_python_v1.1_src.tar.gz** from our web site:

http://www.stsci.edu/resources/software_hardware/pyraf/current/download

Unpack the tar file. This will create a directory stsci_python. Installing the packages is performed in two steps - installing Numarray and installing all other packages and modules.

4.1 Installing PyRAF, PyFITS, Readgeis and Numdisplay

4.1.1 Installation in Python's site-packages.

Go down in the stsci_python directory and build PyRAF and all other python modules:

1. cd stsci_python
2. python setup.py install

This will install pyraf, pyfits, readgeis, fitsdiff and numdisplay in the default location - python's site-packages directory. Depending on where Python is installed, you may need root write permission to do this installation. The PyRAF start up script will be located with the python executable, which most probably is already on your $PATH.

4.1.2 Personal Installation.

If you don't have root privilege or want to do a personal installation, this can be done by executing one of the sets of commands below. In all these cases PYTHONPATH must be modified for Python to be able to import the modules. For each case we give a specific example for installing in a directory `/home/user/myinstall.`

- Using the option “--home=<install-dir>”

  The packages will be installed under <install-dir>/lib/python. The PyRAF start up script will be in <install-dir>/bin.

```
python setup.py install --home=<install-dir>

setenv PYTHONPATH <install-dir>/lib/python
```

Example:

%cd stsci_python

%python setup.py install --home=/home/user/myinstall

%setenv PYTHONPATH /home/user/myinstall/lib/python

To start PyRAF:

%/home/user/myinstall/bin/pyraf

- Using the option “--prefix=<install-dir>”

  This installs the packages under <install-dir>/lib/python2.3/site-packages

```
python setup.py install --prefix=<install-dir>

setenv PYTHONPATH <install-dir>/lib/python2.3/site-packages
```

Example:

%cd stsci_python

%python setup.py install --prefix=/home/user/myinstall

%setenv PYTHONPATH /home/user/myinstall/lib/python2.3/site-packages

To start PyRAF:

%/home/user/myinstall/bin/pyraf

- To install directly in <install-dir>, PyRAF, Numdisplay and the individual modules in stsci_python have to be install one at a time. The PyRAF start up script will be located in <install-dir>/pyraf/.

```
cd stsci_python/pyraf

python setup.py install --local=<install-dir>

cd stsci_python/numdisplay

python setup.py install --install-lib=<install-dir> --install-data=<install-dir>/
numdisplay
```

To install all other Python modules resident in stsci_python, just copy them to <install-dir>:

```
cd stsci_python

cp *.py <install-dir>

setenv PYTHONPATH <install-dir>
```

Example:
%cd stsci_python/pyraf
%python setup.py install --local=/home/user/myinstall
%cd stsci_python/numdisplay
%python setup.py install --install-lib=/home/user/myinstall --install-data=/home/user/myinstall/numdisplay
%cd stsci_python
*%cp *.py /home/user/myinstall*
setenv PYTHONPATH /home/user/myinstall
To start PyRAF:
%/home/user/myinstall/pyraf/pyraf

4.2 Installing Numarray

4.2.1 Installing in Python's `site-packages`:

To install Numarray execute the following commands:

```
% cd stsci_python/numarray-<version>

% python setupall.py install --gencode
```

This will install numarray in the default location - Python's `site-packages` directory.

Example:
%cd stsci_python/numarray-0.6.2
%python setupall.py install --gencode

4.2.2 Personal Installation

To install in a personal directory, use

```
% setenv PYTHONPATH <install_dir>

% python setupall.py install --gencode --local=<install-dir>
```

Example:

```
%cd stsci_python/numarray-0.6.2

setenv PYTHONPATH /home/user/myinstall/

%python setupall.py install --gencode  --local=/home/user/myin-
  stall/
```

Note, that in order to install all packages in Numarray (which is required for some STSDAS tasks), the `setupall.py` script has to be used. This will not be necessary starting with Numarray v0.8.

5.0 PyRAF Setup

The default installation installs the `pyraf` startup script in the same directory with the python executable, so this directory has to be on the search path. If the software was installed in a personal directory, with either `“--home=”` or `“--prefix=”,` then the startup script is located in `<install-dir>/bin.` In order to run PyRAF conveniently, you can define an alias or create a symbolic link to the startup script.

```
 alias pyraf <install_dir>/bin/pyraf

 cd /usr/local/bin (or some other directory on PATH)

ln -s <install-dir>/bin/pyraf pyraf
```

6.0 Installing PyRAF with the Opengl graphics kernel

PyRAF supports two different graphic kernels, based on Tkinter and OpenGL, the former being the default.

To be able to use the opengl kernel, you need to have Mesa installed on your system. (PyRAF does not work with other OpenGL implementations.) We have noticed that PyRAF has best performance with Mesa v3.4.2. If you install Mesa, make sure that its libraries are on the LD_LIBRARY_PATH if necessary.

The following command will install PyRAF with support for the opengl kernel.

```
% python setup.py install --with-opengl=<opengl-libraries-dir>
```

Please, note, that generally this installation is more difficult than the default one.

7.0 Assistance

If you have any difficulties with the installation of any of the packages in `stsci_python`, **please** do not hesitate to contact us for assistance. Also, if you just have questions or suggestions, contact us at *help@stsci.edu.*