STSCI_PYTHON v2.2

Source Release

Installation Instructions

Astronomy Tools And Applications
Space Telescope Science Institute

Stsci_python is a collection of Python modules, packages and C extensions that has been developed to provide a general astronomical data analysis infrastructure. They can be used for Python tasks that are accessible from within STSDAS when running under PyRAF or standalone from within Python.

Version 2.2 of stsci_python contains new packages and updated versions of some of the old packages.Please consult section “What’s in stsci_python_2.2” for a complete list of all packages.

Before attempting an installation, please make sure you have all required supporting packages installed and working. If you have experience doing installations, the “Quick Installation Directions” is probably a good place to start. In case of problems, consult the detailed explanation of the installation process.

1.0 What’s in stsci_python v2.2

1.1 New Packages and Modules

- Iterfile   v 0.1
- Saaclean v0.5

1.2 Updated Packages and Modules

- Numarray v1.4.1
- Pyfits v1.0
- PyRAF v1.2
- Numcombine v0.3
- Nimageiter v0.5
- Imagestats v1.0
- Multidrizzle v2.7
- Pydrizzle v5.6
- Parseinput v0.1.5
- Readgeis v1.8

1.3 Unchanged Packages

- Fitsdiff v 1.3
- Imageiter v0.1
- Irafglob v1.0
- Numdisplay v1.1
- Readgeis v1.7
- Makewcs v0.7

2.0 Required Supporting Packages

As a convenience all supporting packages are available from

`http://www.stsci.edu/resources/software_hardware/pyraf/support_stsci_python`

If you haven't installed all supporting packages, we suggest that you follow the order in which they are listed below:

- IRAF
- TABLES/STSDAS - 3.4
- Tcl/Tk - v8.3 or higher
- Readline
- Python - v2.3 or higher (with readline and _tkinter modules enabled)
- Numeric
- libf2c (**Note:** the package on our web site has a change that allows it to work on 64bit Linux platforms)

- Pmw 1.2

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

```
%python

>>>import readline

>>>import Tkinter

>>>Tkinter._test()

>>>import Numeric

>>>import Pmw
```

If you don’t get an ImportError, this means that Tcl, Tk, Readline, Python, Numeric and Pmw are already installed on the system.

On MacOSX the installation was tested with the supporting packages installed both through fink 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.

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. Follow the directions on their site to get IRAF working on your system.

(http://iraf.noao.edu)

3.0 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.

If you have a previous installation of stsci_python, you will need to clean the old versions. Please read "Cleaning an old installation" for a list of packages that need to be removed.

Unpack the source distribution file somewhere and build Numarray.

Note: **Numarray must be the first package to build because it's needed by some of the other packages.**

Note: **The default installation directory of Python packages is** `python-<version-number>/site-packages.` **The installation directions in this section will do a default installation in Python's** `site-packages.` **It is possible to install in a different directory - read section "Personal Installation" or type "**`python setup.py --help`**" to find out how to do that.**

```
%gunzip stsci_python_2.2.tar.gz

%tar -xvf stsci_python_2.3.tar

%cd stsci_python

%cd numarray-v1.3.3

%python setup.py config install --gencode
```

This will install numarray in Python's site-packages directory. To install the rest of the packages, type:

```
%cd ../

%python setup.py install --with-f2c=<location-of-f2c>,
```

where <location-of-f2c> is the path to the directory with `libf2c.a` and `f2c.h`.

4.0 Cleaning An Old Installation

This release has updated versions of almost all packages so we recommend removing or renaming all of them before attempting an installation of v2.2. Make sure you remove all relevant directories, numarray for example installs in two directories by default:

/usr/local/Python-2.3.4/lib/python2.3/site-packages/numarray/

/usr/local/Python-2.3.4/include/python2.3/numarray/

5.0 Installing The Supporting Packages

Some of the packages require IRAF to be present on the system. 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`**.

Note: A full installation of the supporting packages is needed, including **libraries** and **header files**. On some operating systems the header files may be in a separate package. For example on Redhat they are in the corresponding “devel” rpm package.

Note for MacOS X users #1: Iraf and stsci_python need X11 and will not work with the version of python distributed with the operating system (`/usr/bin/python`) or with macpython.

Note for MacOS X users #2: All supporting packages can be installed with fink on MAcOSX. Our experience shows that this is generally easier. 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.

Note: Python **MUST** be installed from source using fink, in order to get Tkinter working. If fink was used for installations of all supporting packages on MacOSX, make sure you are running python in /sw/bin when you attempt to install stsci_python.

Note for Linux users: The packaging systems for the various flavours of Linux can be used to install the supporting packages. For example on Redhat rpms can be used for all supporting packages. However, Tkinter rpms should be matched with the version of python and the operating system.

Source Installations on the command line: 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>`. The option '`--prefix=`' in the above '`./configure`' command may be omitted for installations in /usr/local.

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
```

5.1 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
```

5.2 Readline

Readline is probably already installed on your system. Make sure the header files are installed as well. A possible location to look for them is /usr/include/readline. In case you need to install Readline in your personal directories, the following commands will install it on most systems:

```
%cd readline

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

%make

%make install
```

5.3 Python

Linux: Some versions of Linux still come with Python 2.2 installed. Since Python2.3 is required, you will need to install it from rpms or from source. If rpms are installed, usually Tkinter is distributed separately in a different rpm package. Make sure the Tkinter rpm package was built for this version of Python and for this operating system.

Source Installation: Python is available from http://www.python.org. 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>

%make

%make install
```

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

To build Tkinter, 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
```

5.4 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.`

5.5 Pmw

To install Pmw, unpack the tar file and copy the directory **Pmw** to your installation directory (for example Python's `site-packages/`).

5.6 Libf2c

Note: The package on our web site has support for 64 bit Linux.

If a source installation is needed, download the package from our web site. It is needed only at compile time but it is worth installing it permanently on the system for future upgrades of stsci_python. The following commands will build it:

```
%tar xvf libf2c.tar

%cd libf2c

%make -f makefile.os
```

Substitute makefile.os with the correct makefile for your operating system. For example for 64 bit linux operating system use:

```
%make -f makefile.linux64
```

After building libf2c the two files that are needed, `f2c.h` and `libf2c.a,` will be located in the source directory. They can be moved to a permanent location.

6.0 Installing stsci_python

Download the file `stsci_python_2.2.tar.gz` from our web site:

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

Unpack the tar file somewhere (it can be removed after the installation is completed). This will create a directory `stsci_python.` Installation is performed in two steps - installing `Numarray` and installing all other packages and modules.

Note: **Numarray must be installed first!**

Note: In the discussion below, `<install-dir>` stands for the full path to the directory in which you wish to install the packages.

6.1 Numarray Installation

Before you proceed to install Numarray, make sure there is no previous installation of Numarray in the installation directory.

6.1.1 Default Installation In Python's site-packages

To install Numarray execute the following commands:

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

% python setup.py config install --gencode
```

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

Example:

```
%cd stsci_python/numarray-1.3.3
%python setup.py config install --gencode
```

6.1.2 Personal Installation

To install in a personal directory, use

```
  % setenv PYTHONPATH <install_dir>
% python setup.py config install --gencode --local=<install-
  dir>
```

Example:

```
%cd stsci_python/numarray-1.3.3
setenv PYTHONPATH /home/user/myinstall/
%python setup.py config install --gencode  --local=/home/user/myinstall/
```

6.2 Installing All Other Packages in stsci_python

6.2.1 Default Installation in Python's site-packages

Go down in the stsci_python directory and build and install all packages and modules:

```
%cd stsci_python

%python setup.py install --with-f2c=<f2c-dir> ,
```

where <f2c-dir> is the path to the directory with `libf2c.a` and `f2c.h.`

This will install all packages and modules listed in section "*What's in stsci_python_2.2*" 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.

6.2.2 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.` f2c is installed in `stsci_python/f2c.`

Note: Numarray must be on your PYTHONPATH before attempting to install the other packages. Generally it will not be on PYTHONPATH if it was installed in a personal directory.

The examples below assume that numarray was installed with the same option and is in <install-dir>.

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

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

```
              %setenv PYTHONPATH <install-dir>
python setup.py install --with-f2c=<f2c-dir> --local=<install-dir>
              setenv PYTHONPATH <install-dir>
```

  Example:

  %cd stsci_python

  %setenv PYTHONPATH /home/user/myinstall/

  %python setup.py install with-f2c=./libf2c --local=/home/user/myinstall

  To start PyRAF:

  %/home/user/myinstall/pyraf/pyraf

- 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.

```
              %setenv PYTHONPATH <install-dir>/lib/python
python setup.py install --with-f2c=<f2c-dir> --home=<install-dir>
              setenv PYTHONPATH <install-dir>/lib/python
```

  Example:

  %cd stsci_python

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

  %python setup.py install with-f2c=./libf2c --home=/home/user/myinstall

  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

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

```
python setup.py install --with-f2c=<f2c-dir> --prefix=<install-dir>

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

Example:

%cd stsci_python

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

%python setup.py install --with-f2c=./libf2c --prefix=/home/user/myinstall

To start PyRAF:

%/home/user/myinstall/bin/pyraf

7.0 PyRAF Setup

After a default installation the `pyraf` startup script is located in the same directory with the python executable, so this directory has to be on the search path. Modify the environment variable $PATH if necessary. If the software was installed in a personal directory, with either `“--home=”` or `“--prefix=”`, then the startup script is located in `<install-dir>/bin`. If `“--local=”`, then pyraf’s startup script is in `<install-dir>/pyraf`. 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
```

Note for Linux users: There is a problem running iraf on some Linux systems (Fedora, RHEL, possibly others). Quote from the iraf web site:

“ Specifically, pointers allocated in the normal course of a task may occasionally be at an address outside the user’s per-process stack space, resulting in a "memory has been corrupted" or "segmentation violation" error. The workaround is to remove the stacksize limit in the user’s shell with a command such as:

```
%limit stacksize unlimited          # for tcsh users

%ulimit -s unlimited                # for bash users
```

As a preventive measure against this problem, the CL startup script was modified to implement this change and so most users who only use IRAF from the CL will not need to take any special action."

PyRAF does not use the CL start up script, so you will need to have this command in your unix start up files (for example ~/.login).

8.0 Testing The Installation

The script `stsci_python/test22.py` can be used to check the integrity and version compatibility of the installation.

Note: This script must be run from a directory other than stsci_python, for example your home directory. Make sure that $PYTHONPATH is correct.

```
%cp stsci_python/test22.py ~

%cd

%python test22.py
```

9.0 Installing PyRAF with the Opengl graphics kernel

We are dropping support for the OpenGL graphics kernel in PyRAF.

If someone really MUST use it let us know.

10.0 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.*