JWPSF Version 3.1 Installation Instructions

The program jwpsf is supplied as a Python source. You will therefore need to have Python installed on your system. In addition we rely on several supporting packages, principally numPy, Pyfits and Matplotlib. The other items listed below are normally included with these and so your installation should be simpler than the full instructions below suggest.

This release simplifies the installation and program startup. One way to ensure you have the appropriate versions of Python, numpy, Pyfits and Matplotlib is to install the package for your system supplied by the Space Telescope Science institute at

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

It is possible to run jwpsf without Matplotlib. In this case the calculations will proceed omitting all display graphics. The output will still be a FITS image of the PSF which you can display and analyze with any tools you normally use.

After downloading the tar file jwpsf_v3.1.tar into your working directory extract the files either by clicking on the tar file icon or by issuing the command

```
tar -xvf jwpsf_v3.1.tar.gz
```

followed by

```
cd jwpsf
python setup.py install
```

Users at STScI and possibly at other institutes may encounter permission problems because the installation attempts to write into usr/ If so you will need to set up your own bin directory and paths with the following lines

```
setenv PYTHONPATH ~/py/lib/python:$PYTHONPATH
setenv PATH ~/py/bin:$PATH
```

and then install with

```
 tar -xvf jwpsf_v3.1.tar.gz
cd jwpsf
python setup.py install --home ~/py
```

The first two setenv lines will need to be placed in your system startup commands so that these paths will be in effect the next time you run the program. You can now start the program by issuing the commands

```
python
import jwpsf
jwpsf.jwpsf()
```

or more simply with

runjwpsf.py

which will run a script containing the above commands. You no longer need be in any particular directory to run the program. The only consideration is that a Results directory will be created in your working directory where the output FITS files are, by default, collected. So always operating from the jwpsf directory will maintain all results in the same place. The Results directory may be treated as a scratch directory. You are free to delete or move files from here

If your program is running you need read no further.

Required Supporting Packages

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

- Tcl and Tk - v 8.3 or later
- readline
- Python – v 2.5, with Tkinter and readline enabled
- Numpy – v 1.0.4

  http://sourceforge.net/projects/numpy/
- pyfits – v 1.3

  http://www.stsci.edu/resources/software_hardware/pyfits
- freetype, libpng, zlib (if needed)
- matplotlib – v 0.91.2

  http://www.matplotlib.sf.netTo install Python from source, Tcl, Tk and Readline are needed, including header files and libraries. (**Note:** Header files on some linux distributions are in the corresponding `devel` package.) Matplotlib installations require zlib, freetype, and libpng. (**Note:** Psf can be run without matplotlib.)

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 numpy

>>>import pyfits
```

```
>>>import matplotlib
```

If you don't get an ImportError, this means that all necessary packages are already installed on the system.

On MacOSX the installations can be done either through fink or from source on the Unix command line. We find that generally installations using fink are easier. All supporting packages can be installed through fink. On MacOSX we have tested only with X11 based Tkinter. This requires the X11 based Tcl and Tk. Although untested Psf should work with macpython. For some of the external dependencies, a C compiler (and optionally the X11 windowing system, if you are using X11 based Tk and Python) are needed. On MacOSX compilers are available with the Developer's tools.

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 PSF, you will need to clean the old version first.

```
%gunzip psf_v1.tar.gz

%tar -xvf psf_v1.tar

%cd psf_v1

%python setup.py install
```

Cleaning An Old Installation

Before installing a new version of Psf, remove all files from the old installation. If a standard installation was performed all files are in Python's site-packages directory.

Installing The Supporting Packages

Source Installations on the command line: If installation from source is necessary, on most systems for most packages 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 `--prefix` is not given on the command line, by default all packages will be installed 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
```

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 (the header files may be in a separate 'devel' package).

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
```

Readline

Readline is probably already installed on your system. Make sure the header files are installed as well. 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
```

Python

Linux: Tkinter is distributed separately in a different rpm package. If you are installing rpms, make sure the Tkinter rpms match the version of Python and the version of the operating system you are using.

Source Installation: If you need to install from source, download the source file 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>
```

Enable Tkinter and readline in Modules/Setup as described below and build python:

```
%make

%make install
```

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 commented out 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
```

NumPy

NumPy is available from:

`http://numpy.scipy.org/`

To install NumPy execute the following commands:

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

% python setup.py install
```

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

To install in a personal directory, use the commands below and change PYTHONPATH to point to the installation directory:

```
% python setup.py config
   % python setup.py install –install-lib=<install-dir> --
   install-scripts=<install-dir>/numpy
```

Example:

%cd stsci_python/numpy-1.0.4

%setenv PYTHONPATH /home/user/myinstall/

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

PyFITS

PyFITS is available from:

http://www.stsci.edu/resources/software_hardware/pyfits

Download the tar file and unpack it. To install it in site-packages:

```
%cd pyfits-1.0

% python setup.py install
```

To install in a personal directory, use the commands below and change PYTHONPATH to point to the installation directory:

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

Matplotlib

Matplotlib depends on several external libraries - zlib, freetype2, libpng and NumPy Most of them (possibly all of them) are already present on your system. If not below are brief instructions on how to compile these packages. Please consult the documentation in the package directories if you have difficulties. If you need to install any of the dependencies, it may be practical to install all of them in one directory (for example /usr/local, if you have access there, or a personal directory).

Zlib

Zlib is available from

http://www.zlib.net/

It is included in many operating systems, check for the presence of libz in the system directories. If not present, look at the web site for binaries or for instructions to compile from source.

Freetype2

Freetype2 is available at

http://freetype.sourceforge.net/index2.html

Note that Freetype2 is needed, not Freetype1. Many operating systems come with a freetype library, so there's a good chance it is already available. Matplotlib works with v 2.1.7 or later and you may need to install a later version. Since updating the existing freetype may brake window managers, it's best to keep the one that comes with the system and install the new version in a different directory. Freetype2 installation uses the GNU autoconfig and make system. In most cases the following commands will install the library:

```
%./configure

%make

%make install
```

If you are not on a GNU system, you may need to use gmake explicitely. If there's a problem with the installation, consult the package installation directions.

Libpng

Check with your sys admin if libpng is already available. It can be downloaded from

http://www.libpng.org/pub/png/

Follow the installation instructions that come with the package to install it.

Matplotlib

Matplotlib's web site is http://matplotlib.sourceforge.net . In case of installation problems it's best to look at the web site and the users mailing list. There are also links to precompiled binaries.

Note for MacOSX users: Fink can be used to install matplotlib, but the fink version of matplotlib was compiled with macpython. If you are using fink, make sure you meet all the dependencies required by that package.

If a source installation is required and all dependencies are installed in system directories, matplotlib can be installed in python's site-packages with the command:

%python setup.py install

This may or may not require root access depending on where python is installed. If any of the extrenal dependencies was installed in a non-system directory, you will need to edit matpotlib/setupext.py and add that directory to your platform entry in basedir.

Installing Psf

System wide installation

To install in python's site-packages, execute:

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

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 "--local=<install-dir>"

  The packages will be installed under <install-dir>.

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

Example:

%cd psf

%setenv PYTHONPATH /home/user/myinstall/

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

- Using the option "--home=<install-dir>"

  The packages will be installed under <install-dir>/lib/python.

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

Example:

%cd psf

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

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

- 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 psf

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

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

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.*