The PESSTO Pipeline – installation and testing instructions

S. Valenti, S. J. Smartt, K. Smith, E. Cappellaro, M. Fraser

There are two parts to this installation guide. First, carry out the installation on your particular platform. We have currently tested it independently on Snow Leopard (Mac OSX10.6.8), Lion (Mac OSX 10.7.5), Mountain Lion (Mac OSX 10.8.2) and Ubuntu 12.04. S. Valenti has also installed it on Linux CENTOS and ubuntu 10.10 platforms, but this has not yet been independently tested. Secondly, carry out two tests to check the installation works.

The latest version of the pipeline : is ntt.1.1.2 (29th August 2012). This document may refer to the ntt.1.0 version in places, obviously you should use the latest version and change the installation commands appropriately.

1. Installation

The PESSTO pipeline is a pyraf and python based code which employs the familiar IRAF spectroscopic reduction routines (i.e. ccd processing, extraction, wavelength and flux calibration). It requires python to be installed and python 4 packages (along with Sextractor) all of which are distributed with Scisoft. This manual guides you through the installation and troubleshoots some known problems.

Snow Leopard Mac OSX 10.6.8

Step 1 – Check of your python installation

The PESSTO pipeline is a *python* based code and requires the following *python* modules :
pyraf numpy pyfits pylab

To check if these work on your machine, go to the dir in which you have installed IRAF and have a login.cl file

```
/Users/sjs>python
Python 2.7.1 (r271:86832, Dec 30 2010, 17:12:17)
[GCC 4.2.1 (Apple Inc. build 5664)] on arwin
Type “help”, “copyright”, “credits” or “license” for more information.
>>> import numpy
>>> import pyfits
>>> import pylab
>>> import pyraf
```

If you get no errors, then these modules are installed. This is where you want to be before you install the pipeline itself. If you have no errors at this point, go to step 3.

Step 2 – Errors and fixes for your python installation

If you get any error messages then first of all try this :

If you are a ***scisoft*** user, then source the setup files and try the python check again

```
source /usr/local/scisoft/bin/Setup.csh  (if you are a tcshell user)
source /usr/local/scisoft/bin/Setup.bash (if you are a bash user)
```

It's quite likely that this will <u>not</u> fix the problem and you will still get a "Bus Error" or "Segmentation Errror", when importing ***pylab*** .

You most probably have ***pyraf*** installed within your ***scisoft*** installation. We think there is a problem with the ***scisoft*** version under Snow Leoapard and don't recommend that you try and fix it. At least we haven't pursued that route or tried it.

Download and install the STScI IRAF package, which includes ***python*** modules that we know work
http://stsdas.stsci.edu/download/stsci_python_2.12/iraf.intel.snow_leopard.dmg

This creates the dir

/usr/stsci

And installs IRAF and python (currently Python-2.7)

Now you need to specify which ***python*** installation you want to work, as you may now have two installed. This depends on whether you use ***bash*** or **tcsh**

<u>tcsh user instructions :</u> If you are a tcsh user then add these TWO lines to your .tcshrc (or .cshrc) file in your home directory

```
if ( -f /usr/stsci/envconfig.mac/cshrc )      source /usr/stsci/envconfig.mac/cshrc
iraf
```

Of course you must then comment out the sourcing of the scisoft Setup.csh file, which is likely in your .tcshrc or .cshrc file, add a ' ***#*** ' to the beginning of the line

```
# source /Applications/scisoft/all/bin/Setup.csh
```

Now check you are pointing at the right python path, by either opening a new Terminal window or sourcing your .tcshrc file.

```
/Users/sjs>source ~/.tcshrc

/Users/sjs>which python
/usr/stsci/pyssg/Python-2.7/bin/python
```

Also, if you have a directory called *.matplotlibrc* in your home directory, then delete it now.

All should be OK – now check again that you can import **pyraf numpy pyfits pylab**

```
/Users/sjs>python
Python 2.7.1 (r271:86832, Dec 30 2010, 17:12:17)
[GCC 4.2.1 (Apple Inc. build 5664)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import pyfits
>>> import pylab
>>> import pyraf
```

Yes ? …. Good to move to next step

No ? ….. Most likely you are not pointing to the STScI version of python. Go back and check your ***.tcshrc*** file and where it is sourcing python from.

bash user instructions : There are none – this only works for T/C shell, because there is only a csh environment file provided with the STScI installation of IRAF/prfay/python.

Step 3 – Check of your SExtractor, swarp and wget installation

The pipeline requires that you have the SExtractor, swarp and wget binaries pre-installed.

You will also need to have *SExtractor*[1] installed. To check this, type:

```
/Users/sjs>which sex
/usr/bin/sex
```

In this case, it's installed in /usr/bin. If it is installed, you are good to go to Step 4.

If not, then you can install with the instructions on the SExtractor website or just download the binary for Snow Leopard, available below, and copy into a bin directory (which obviously must be in your path).

[1] http://www.astromatic.net/software/sextractor

Snow Leopard sextractor binary:
[***http://www.pessto.org/pessto/private/pipeline***](http://www.pessto.org/pessto/private/pipeline)

After getting the binary, copy it to /usr/bin (you will probably need to use sudo, due to root privileges on this dir)

```
/Users/sjs>sudo cp sex /usr/bin
```

Similarly, the pipeline requires *swarp*[2] to be installed. You can install with instructions on the *swarp* website, or just download the binary for Snow Leopard, available below, and copy into a bin directory as before.

First check if it's there

```
/Users/sjs>which swarp
/usr/bin/swarp
```

And if not, then download from :
[***http://www.pessto.org/pessto/private/pipeline***](http://www.pessto.org/pessto/private/pipeline)

and as before :

```
/Users/sjs>sudo cp swarp /usr/bin
```

Finally, you will need the *wget*[3] binary, which you can either download, compile and install yourself or download a precompiled binary from:
[***http://www.pessto.org/pessto/private/pipeline***](http://www.pessto.org/pessto/private/pipeline)
If you download the binary, please place move it to /usr/bin or /usr/local/bin as the superuser (and make sure its permissions are executable - i.e. chmod 755 wget).

Step 4 – Install the PESSTO pipeline

Now download the PESSTO pipeline source code from here:
[***http://www.pessto.org/pessto/private/pipeline***](http://www.pessto.org/pessto/private/pipeline)

untar the pipeline

```
/Users/sjs>tar -zxvf ntt-1.0.tar.gz
ntt-1.0/
ntt-1.0/bin/
…..
….
```

[2] http://www.astromatic.net/software/swarp

[3] http://www.gnu.org/software/wget/

Change to the directory created by the unpacking

```
/Users/sjs>cd ntt-1.0
/Users/sjs>pwd
/Users/sjs/ntt-1.0
```

The setup.py file which runs the installation requires root permissions, so you need to open a root shell.

```
/Users/sjs/ntt-1.0> sudo tcsh
```
<enter password if prompted>

You are now logged in as root, and you can check by doing

```
/Users/sjs/ntt-1.0> whoami
root
```

Now run the python setup file (in your root shell)

```
/Users/sjs/ntt-1.0> python setup.py install
running install
running build
running build_py
creating build
creating build/lib
creating build/lib/ntt
....
...
```

Alternatively, you can install the pipeline (if you are not root) in your user directory using the option --prefix=/home/user/xxx

this option will create the following directories inside /home/user/xxx/

/home/user/xxx/bin/

/home/user/xxx/lib/

/home/user/xxx/lib/python2.x/

/home/user/xxx/lib/python2.x/site-packages/

```
/Users/sjs/ntt-1.0> python setup.py install --prefix=/home/user/xxx
running install
running build
running build_py
creating build
creating build/lib
creating build/lib/ntt
```

….

…

if you install the pipeline as user (not as root), you should add this directory
/home/user/xxx/lib/python2.x/site-packages/
to your PYTHONPATH envirorment:

export PYTHONPATH=$PYTHONPATH:/home/user/xxx/lib/python2.x/site-packages (for bash shell)

setenv PYTHONPATH $PYTHONPATH:/home/user/xxx/lib/python2.x/site-packages (for tcshell)

Now you should have the pipeline installed (in this case, in /usr/stsci/pyssg/Python-2.7/bin).

Now make sure to exit the root shell. Simply type exit :

```
/Users/sjs/ntt-1.0> exit
```

Alternatively, kill this window and open a new Terminal window.

Source your .tcshrc file and check that your shell recognizes the PESSTO executables :

```
/Users/sjs/ntt-1.0>source ~/.tcshrc

/Users/sjs/ntt-1.0>which PESSTO
/usr/stsci/pyssg/Python-2.7/bin/PESSTO
```

You should see the following files in /usr/stsci/pyssg/Python-2.7/bin

```
/Users/sjs/ntt-1.0>ls -lrt /usr/stsci/pyssg/Python-2.7/bin/
….
…
…
-rwxr-xr-x    1 root    kmem    14744 14 May 15:22 PESSTOWISE*
-rwxr-xr-x    1 root    kmem     3812 14 May 15:22 PESSTOSOFIPHOT*
-rwxr-xr-x    1 root    kmem     3912 14 May 15:22 PESSTOSOFI2dSPEC*
-rwxr-xr-x    1 root    kmem     2367 14 May 15:22 PESSTOSOFI1dSPEC*
-rwxr-xr-x    1 root    kmem     1744 14 May 15:22 PESSTOFASTSPEC*
-rwxr-xr-x    1 root    kmem     8437 14 May 15:22 PESSTOEFOSCPHOT*
-rwxr-xr-x    1 root    kmem     5267 14 May 15:22 PESSTOEFOSC2dSPEC*
-rwxr-xr-x    1 root    kmem     2945 14 May 15:22 PESSTOEFOSC1dSPEC*
-rwxr-xr-x    1 root    kmem     1294 14 May 15:22 PESSTO*
```

If you have reached this stage, congratulations, you are now you are ready to run the PESSTO pipeline. Now go to Section 2 “Testing the PESSTO pipeline” to check if your installation runs.

Step 5 (optional) : Re-installation of a new version.

If you have successfully installed a version of the pipeline and want to update to a new version, then you can follow the steps below.

Download the new version of the pipeline, and gunzip and untar it as above in any directory you like. You will first need to uninstall the old version

```
/Users/sjs/pessto/pipeline>which PESSTOFASTSPEC
/usr/stsci/pyssg/Python-2.7/bin/PESSTOFASTSPEC
/Users/sjs/pessto/pipeline>cd /usr/stsci/pyssg/Python-2.7/bin/
```

Now the executable files are likely to be owned by root and you do not have permission to delete, hence

```
/usr/stsci/pyssg/Python-2.7/bin>sudo tcsh
Password: enter your mac user password
/usr/stsci/pyssg/Python-2.7/bin>rm PESSTO*
```

You also need to delete the ntt dir from python site-packages directory. If you don't know were is installed the ntt pipeline, you can start python and type

```
>>> import ntt
>>> ntt.__path__
['/usr/stsci/pyssg/Python-2.7/lib/python2.7/site-packages/ntt']
```

```
/Users/sjs/pessto/pipeline> cd
/usr/stsci/pyssg/Python-2.7/lib/python2.7/site-packages
/usr/stsci/pyssg/Python-2.7/lib/python2.7/site-packages>sudo tcsh
Password:
/usr/stsci/pyssg/Python-2.7/lib/python2.7/site-packages>rm -rf ntt
```

Now you can go to the start of Step 4 and repeat.

Lion (Mac OSX 10.7.5) and Mountain Lion (Mac OSX 10.8.2)

Step 1 – Initial Check of your python installation

The PESSTO pipeline is a *python* based code and requires the following *python* modules :
pyraf numpy pyfits pylab

To check if these work on your machine, do the following :

Go to the dir in which you have installed IRAF and have a login.cl file

```
/Users/kws> python
Python 2.7.1 (r271:86832, Jun 16 2011, 16:59:05)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2335.15.00)] on
darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import pyfits
>>> import pylab
>>> import pyraf
```

If you get no errors, then these modules are installed. This is where you want to be before you install the pipeline itself.

Step 2 - Errors and fixes

If you get any error messages then first of all try this :

If you are a ***scisoft*** user, then source the setup files and try the python check again

`source /usr/local/scisoft/bin/Setup.csh` (if you are a tcshell user)

OR

`source /usr/local/scisoft/bin/Setup.bash`

OR

`. /usr/local/scisoft/bin/Setup.bash` (if you are a bash user)

If this doesn't fix the problem, or you get a "Bus Error" or "Segmentation Errror", when importing *pylab*, try installing this version of Scisoft:

http://www.stsci.edu/~npirzkal/Scisoft/downloads/Intel/10.7.x/

This downloads as a package ready to install. (We currently recommend that you use Scisoft 2012.11.1, but please see notes about manual IRAF startup errors in the Appendix.) After installation the relevant Setup.csh or Setup.bash should be added to the top of your .cshrc, .tcshrc or .bash_profile file. Any existing ones should be commented out.

Comment out the following, if it exists :

`# source /Applications/scisoft/all/bin/Setup.csh` (T/C shell)

or

`# source /Applications/scisoft/all/bin/Setup.bash` (Bash)

and then add either

`source /usr/local/scisoft/bin/Setup.csh` (for T/C shell)

or

`source /usr/local/scisoft/bin/Setup.bash` (for Bash)

Once this is done open a new Terminal window.

All should be OK – now check again that you can import **pyraf numpy pyfits pylab**

```
/Users/kws> python
Python 2.7.1 (r271:86832, Jun 16 2011, 16:59:05)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2335.15.00)] on
darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import pyfits
>>> import pylab
>>> import pyraf
```

Yes ? …. Good to move to next step

No ? ….. Most likely your PYTHONPATH is not setup correctly. Check your PYTHONPATH by typing:

```
echo $PYTHONPATH
/usr/local/scisoft/packages/python/lib/python2.6/site-packages/
```

Now check that the location of PYTHONPATH actually exists. In some recent cases Scisoft has **moved** (correctly) the packages to a directory name containing "python2.7" but the Setup.csh or Setup.bash script is still referring to "python2.6" (around lines 60-65 of the Setup script). If this is the case, edit the Setup.csh or Setup.bash file, re-source the file (see above) and try the python imports again.

Step 3 – Check of your SExtractor installation

You will need to have *SExtractor*[4] installed. To check if you have it type:

```
/Users/kws>which sex
```

"sex" is the SExtractor executable. Hopefully you will see something like this (depending on where the sex is located in your installation :

```
/usr/local/scisoft///bin/sex
```

The Scisoft version for Lion *should* have the SExtractor package and it should install it for your as discussed above for http://www.stsci.edu/~npirzkal/Scisoft/downloads/Intel/10.7.x/

There should not be any requirement to install SExtractor separately.

However if you are having problems, you can install with the instructions on the SExtractor website. Or you could get the executable directly from another Lion user (this has not yet been tested, but we believe it should work).

Similarly, the pipeline requires *swarp*[5] to be installed. The release which is bundled with Scisoft 2012.2.1 and 2012.11.1 is NOT compatible and must be renamed. You can install with instructions on the *swarp* website, or just download the binary for Lion, available below, and copy into a bin directory as before.

First check if there's an existing installation

```
/Users/sjs>which swarp
/usr/local/scisoft/bin/swarp
```

[4] http://www.astromatic.net/software/sextractor

[5] http://www.astromatic.net/software/swarp

If you have the scisoft version please rename the binary:

```
/Users/sjs> sudo mv /usr/local/scisoft/bin/swarp /usr/local/scisoft/bin/swarp_scisoft
```

If you don't have the right version of swarp, then download from :
http://www.pessto.org/pessto/private/pipeline

and as before :

```
/Users/sjs>sudo cp swarp /usr/bin
```

Finally, you will need the *wget*[6] binary, which you can either download, compile and install yourself or download a precompiled binary from:
http://www.pessto.org/pessto/private/pipeline
If you download the binary, please place move it to /usr/bin or /usr/local/bin as the superuser (and make sure its permissions are executable - i.e. chmod 755 wget).

Step 4 – Install the PESSTO Pipeline

Now download the PESSTO pipeline source code :
http://www.pessto.org/pessto/private/pipeline

untar the pipeline. It doesn't matter where to – you could do this in your home directory (as in the Snow Leopard example) or in /tmp.

```
/tmp> tar -zxvf ntt-1.0.tar.gz
ntt-1.0/
ntt-1.0/bin/
…..
….
```

Change to the directory created by the unpacking

```
/tmp> cd ntt-1.0
/tmp/ntt-1.0> pwd
/tmp/ntt-1.0
```

The setup.py file runs the installation. However, this requires root access, because the pipeline package will be installed under the relevant Python site-packages directory.

```
/tmp/ntt-1.0> sudo tcsh
```
<enter password if prompted>

[6] http://www.gnu.org/software/wget/

or

```
/tmp/ntt-1.0> sudo bash
```
<enter password if prompted>

Because you are now the root user, you need to manually source the relevant Scisoft Setup script:

```
/tmp/ntt-1.0> source /usr/local/scisoft/bin/Setup.csh
```

or

```
/tmp/ntt-1.0> source /usr/local/scisoft/bin/Setup.bash
```

Now run the python setup file (should be exactly the same command for both T/C and Bash shells).

```
/tmp/ntt-1.0> python setup.py install
running install
running build
running build_py
creating build
creating build/lib
creating build/lib/ntt
....
...
```

Once the installation is complete, EXIT the root bash shell, returning to your own user ID. (Or start a new Terminal window to make sure.)

```
/tmp/ntt-1.0> exit
```

Now you should have the pipeline installed. In this case it's installed in /usr/local/bin/. Make sure you source your .tcshrc (for tcsh users) or .bash (for bash users) to refresh your paths within the shell.

```
/tmp/ntt/ntt-1.0>source ~/.tcshrc
```
or

```
/tmp/ntt/ntt-1.0>source ~/.bash
```
(or source your equivalent bash setup file).

```
/tmp/ntt-1.0> which PESSTO
/usr/local/bin/PESSTO
```

You should see the following files in /usr/local/bin

```
/tmp/ntt-1.0> ls -l /usr/local/bin | grep PESSTO
-rwxr-xr-x  1 root  wheel      1271 14 May 15:16 PESSTO
-rwxr-xr-x  1 root  wheel      2922 14 May 15:16 PESSTOEFOSC1dSPEC
-rwxr-xr-x  1 root  wheel      5244 14 May 15:16 PESSTOEFOSC2dSPEC
-rwxr-xr-x  1 root  wheel      8414 14 May 15:16 PESSTOEFOSCPHOT
-rwxr-xr-x  1 root  wheel      1721 14 May 15:16 PESSTOFASTSPEC
-rwxr-xr-x  1 root  wheel      2344 14 May 15:16 PESSTOSOFI1dSPEC
-rwxr-xr-x  1 root  wheel      3889 14 May 15:16 PESSTOSOFI2dSPEC
-rwxr-xr-x  1 root  wheel      3789 14 May 15:16 PESSTOSOFIPHOT
-rwxr-xr-x  1 root  wheel     14721 14 May 15:16 PESSTOWISE
```

If you have reached this stage, congratulations, you are now you are ready to run the PESSTO pipeline Now go to Section 2 “Testing the PESSTO pipeline” to check if your installation runs.

Linux CENTOS

This section is not yet an easy installation guide – it needs to be tested and updated as someone walks through the installation and tidies up what is needs to be done.

Login as root user and carry out the following :

```
$ root
$ yum upgrade
$ yum install gcc
$ yum install gcc-gfortran
$ yum install emacs
$ yum install gimp
$ yum install readline-devel
$ yum install svn
$ cd /usr/lib/ su
$ ln -s libreadline.so libreadline.so.5
```

command for the virtual machine

Mount VirtualBox Guest Additions from Devices

add your home as shared directory (auto mount, make permanent)

add pessto user to the group vboxsf (users and groups)

```
$ reboot
```

start the network

Become root user

```
$ su root
$ yum install kernel-devel kernel-headers
$ KERN_DIR=/usr/src/kernels/'uname -r'-'uname -m'
$ export KERN_DIR
$cd /media/VBOXADDITIONS_4.1.12_77245/
$./VBoxLinuxAdditions.run
$ reboot
```

```
##########################################
#####  start the network
```

Become the root user

```
$ su root
$ cd /
$ wget
ftp://ftp.eso.org/scisoft/scisoft7.7/linux/fedora11/tar/scisoft-7.7.0.tar.g
z
$ tar -zxvf scisoft-7.7.0.tar.gz
```

Now exit from root user

```
$ exit
```

Add these to line to your `.bashrc`

```
export SCISOFT_IGNORE_SELINUX_WARNING=1
source /scisoft/bin/Setup.bash
```

```
$ cd /home/pessto/
$ source /home/pessto/.bashrc
$ mkdir iraf
$ cd iraf
```

Now do a `mkiraf` to create a `login.cl, uparm` dir and set your terminal to xgterm

```
$ mkiraf
```

```
$ cd /home/pessto/
$ mkdir tmp
$ cd tmp
```

Become the root user again

```
$ su root
$ wget http://www.pessto.org/pessto/private/pipeline/ntt-1.0.tar.gz
$ tar -zxvf ntt-1.0.tar.gz
$ cd ntt-1.0
$ python setup.py install
$ cd /home/pssto/tmp/
$ rm -rf ntt-*
```

Exit from root user

```
$ exit
```

Linux Ubuntu 12.04

Starting from a machine running Ubuntu 12.04 (32-bit), with no pre-existing iraf install.

Go to

http://www.astro.uson.mx/favilac/downloads/ubuntu-iraf/iso/IRAF_Ubuntu.iso

and download the iso file for IRAF_Ubuntu. Become root with

```
$ sudo -s
```

and make a temporary directory and mount the iso you just downloaded there

```
# mkdir /home/morgan/iso_mount
# cd ~/iso_mount
# mount -t 9660 -o loop /home/morgan/Downloads/IRAF_Ubuntu.iso /home/morgan/iso_mount.
```

Now install IRAF_ubuntu

```
# sh install.sh
```

When the script finishes, logout from root

```
# exit
```

and make a directory to keep your iraf login file

```
$ cd ~
$ mkdir iraf
```

Go to the new iraf directory and run mkiraf

```
$ sudo mkiraf
```

Select xgterm as your terminal when prompted. Next, you need to edit the login.cl file which has created, adding in "onedspec" as a package to be loaded at startup. Open ~/iraf/login.cl with your text editor du jour, and add in "onedspec" so that the last few lines look as follows:

```
tv                         # image display
```

```
utilities         # miscellaneous utilities
noao              # optical astronomy packages
onedspec

keep
```

When done, save and exit your text editor.

Next you will likely need to change the permissions and ownership of the ~/iraf directory which was created (obviously use your own username here).

```
$ sudo chown -R morgan:morgan iraf/
```

Finally, test everything with the following commands

```
$ cd ~/iraf
$ iraf
```

which should open a DS9 window, and start an xgterm with iraf. If it all looks ok, then logout, and try the following

```
$ python
Python 2.7.3 (default, Apr 20 2012, 22:44:07)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import pyfits
>>> import pyraf
>>> import pylab
```

If there are no errors, then exit with

```
>>> exit()
```

Now to install sextractor (it does not appear to be present in the IRAF_Ubuntu package)

```
$ sudo apt-get install sextractor
```

Check that sextractor runs

```
$ sextractor
```

The sextractor you installed is invoked with 'sextractor', whereas the pipeline looks for 'sex'. To fix this

```
$ sudo cp /usr/bin/sextractor /usr/bin/sex
```

(or create a link instead)

Download the pipeline (ntt-1.0.tar.gz) from the PESSTO webpages (either through your browser, or via wget as follows)

```
$ cd ~
```

```
        $ wget --http-user=[USERNAME] --http-password=[PASSWORD]
http://www.pessto.org/pessto/private/pipeline/ntt--1.0.tar.gz
```

Uncompress the files

```
        $ tar -xvf ntt-1.0.tar.gz
        $ cd ntt-1.0
```

Switch to root and run the install script

```
        $ sudo -s
        # python setup.py install
```

Test the installation as per section 2 of the PESSTO Pipeline Installation Guide

Finally you will need to install the *Swarp* package. We don't yet have a walk through guide for this, so if you need it go to the *Swarp*[7] *website and follow the instructions.*

[7] http://www.astromatic.net/software/swarp

2. Testing your installation of the PESSTO pipeline

Download all the raw data from April 2012 (1.4GB)

http://www.pessto.org/pessto/private/nightly_tarballs/2012.04/nightly_tarballs.html

Put in any directory, and gunzip and untar. e.g.

```
/Users/sjs/data/pessto/2012april> tar -zxvf 2012.04_all_raw_ntt_data.tar.gz
```

Let’s just make sure your shell sees the path to the pipeline executables :

```
/Users/sjs/data/pessto/2012april>which PESSTOFASTSPEC
/usr/stsci/pyssg/Python-2.7/bin/PESSTOFASTSPEC
```

You also need IRAF startup files in your `~/iraf` directory. This `~/iraf` dir **MUST** exist. Even if you have your `iraf` dir (and associated login.cl) somewhere else, you will need to create a new one specifically with this path. Here are two walk throughs :

1. If you do not have a `~/iraf` directory with a `login.cl` file and `uparm` dir :

```
/Users/sjs>mkdir iraf
/Users/sjs>cd iraf
/Users/sjs>mkiraf
-- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type: xgterm
A new LOGIN.CL file has been created in the current directory.
You may wish to review and edit this file to change the defaults.
```

The pipeline calls the IRAF package `onedspec` and this package is not one of the default packages loaded in a standard `login.cl` file. You need to manually edit the `login.cl` file in the `~/iraf` dir to load this package. Simply open the file in a text editor and add the word `onedspec` to the end of the file, before the line which says `keep`. The bottom of your login.cl file should look like this :

```
tv               # image display
utilities        # miscellaneous utilities
noao             # optical astronomy packages
onedspec

keep
```

2. If you already do have a `~/iraf` dir with a `login.cl` file and `uparm` dir :
Then just make sure that `onedspec` is one of the packages loaded by default. If not, then you need to manually edit the `login.cl` file in the `~/iraf` dir to load this package. Simply open the file in a text editor and add the word `onedspec` to the end of the file, before the line which says `keep`. The bottom of your login.cl file should look like this :

```
tv               # image display
utilities        # miscellaneous utilities
noao             # optical astronomy packages
onedspec

keep
```

Experienced IRAF/pyraf users will know that aslong as they have a `login.cl` file in the `~/iraf` dir then they can point to any `uparm` dir in the `login.cl`

Experienced users can also make a soft link from their own `iraf` dir (wherever that may be) to `~/iraf` so that the pipeline sees the dir where it thinks it should be.

Having all this set up, now test that the pipeline runs. First `gunzip` all the compressed data files in your data dir

```
/Users/sjs/data/pessto/2012april>gunzip *
```

Now ready to try first PESSTO pipeline command

```
/Users/sjs/data/pessto/2012april>PESSTOFASTSPEC
EFOSC.2012-04-12T00:21:13.429.fits

####  image name = EFOSC.2012-04-12T00:21:13.429.fits

EFOSC.2012-04-12T00:21:13.429.fits ->
tLSQ12btn_20120412_Gr13_Free_slit1.0_1.fits
cp: ./arc_20110904_Gr13_Free_slit1.0_2.fits and
arc_20110904_Gr13_Free_slit1.0_2.fits are identical (not copied).

###     check in wavelengh performed ...... spectrum shifted of  -1
Angstrom


###      EXTRACTION USING IRAF TASK APALL
```

```
tLSQ12btn_20120412_Gr13_Free_slit1.0_1_f.fits: LSQ12btn
  Extinction correction applied
  Flux calibration applied

#######################################
### end of reduction
### output files:
tLSQ12btn_20120412_Gr13_Free_slit1.0_1_ex.fits
tLSQ12btn_20120412_Gr13_Free_slit1.0_1_f.fits
tLSQ12btn_20120412_Gr13_Free_slit1.0_1_2df.fits
tLSQ12btn_20120412_Gr13_Free_slit1.0_1_f.asci
```

You will see that the files created are

```
*_ex.fits    ....the extracted 1D spectrum, in ADU
*_f.fits     ....the extracted 1D spectrum, flux calibrated
*_2df.fits   ....the detrended 2D image, flux and wavelength calibrated
*_f.asci     ....ascii version of extracted 1D spectrum, flux calibrated
```

Now also try `PESSTOEFOSCPHOT` as this command requires `SExtractor`. This is a check that your installation locates the executable for `SExtractor`, and that the imaging detrending and astrometric calibration works.

```
/Users/sjs/data/PESSTO/2012april/raw>PESSTOEFOSCPHOT EFOS
C.2012-04-21T01:53:28.893.fits

######################################################################

### input EFOSC.2012-04-21T01:53:28.893.fits snunt121_20120421_B639
### bias  yes masterbias_20120121.fits
### flat  yes flat_20120411_B639.fits
### name  snunt121_20120421_B639_1.fits

### bad pixel mask correction ..... done

### check astrometry: fine
### rmsx rmsy nstars: 0.36 0.328 4

###  time of running 0.0920515139898 minutes

### making a tar with pre-reduced images ........ please wait

### tar file: logfile_phot_apr12_d21to21_efosc_201205221220.tar.gz
```

The following files should be created :

```
-rw-r--r--  1 sjs  staff   3.9M 22 May 12:20 flat_20120411_B639.fits
-rw-r--r--  1 sjs  staff   3.9M 22 May 12:20 masterbias_20120121.fits
-rw-r--r--  1 sjs  staff   127K 22 May 12:20 logfile
-rw-r--r--  1 sjs  staff   2.2K 22 May 12:20 logNTT.txt
-rw-r--r--  1 sjs  staff    19K 22 May 12:20 bad_pixel_mask.pl
-rw-r--r--  1 sjs  staff   3.9M 22 May 12:20 snunt121_20120421_B639_1.fits
-rw-r--r--  1 sjs  staff   3.2M 22 May 12:20
```

```
logfile_phot_apr12_d21to21_efosc_201205221220.tar.gz
-rw-r--r--  1 sjs  staff   864B 22 May 12:20
logfile_phot_apr12_d21to21_efosc_201205221220.pkl
```

`snunt121_20120421_B639_1.fits` is an astrometrically calibrated, detrended image.

If these are created successfully then your installation works, and you can now move on to the Users Manual.

Appendix A – Known issues, problems and work arounds

1. If you have CIAO installed, and the startup script (`/path/to/ciao-4.3/bin/ciao.bash`) is sourced from your `~/.bash`_profile, then you will need to comment this out before installing or running the pipeline.
2. To avoid the pipeline crash because of an incompatibility between matplotlib and pyraf graphic window - two possibilities that Stefano suggests.
   - add this to your .bashrc or .tcshrc :
     export PYRAFGRAPHICS='matplotlib'

   - This seems solve the problem, but slow down your pyraf quite a lot. A better solution is to change your backend of your matplotlib.

     if you have that line in your .bashrc, comment it
     #export PYRAFGRAPHICS='matplotlib'
     add a file in the directory: /Users/yourhome/.matplotlib
     called matplotlibrc
     with the following line in it:

     ```
     backend      : TkAgg
     ```
3. If you have Scisoft 2012.11.1 installed, you will almost certainly not be able to start IRAF manually. This is because there's a bug in one of the bundled IRAF scripts.

   SYMPTOMS:

   ```
   ** Syntax error
   **: if (!defvar("helpdb")) {
                               ^
   **: }
       ^
   ERROR: syntax error
       called as: `cl ()'
       called as: `cl ()'
   Error while reading login.cl file - may need to rebuild with mkiraf
   Fatal startup error.  CL dies.
   ```

   RESOLUTION:
   As the *root* user, edit the following file - e.g.:

   ```
   sudo vim /usr/local/scisoft/packages/iraf/iraf/unix/hlib/extern.pkg
   ```

   Change line 41

   from:

   ```
   if (!defvar("helpdb")) {
   ```

to:

```
if (defvar("helpdb") == no) {
```

(It looks like the exclamation point syntax for negation is no longer recognised.)

4. If you have Scisoft 2012.11.1 installed, you MUST set the ~/.matplotlib/matplotlibrc file contents as stated in point 2 to:

   EITHER

   ```
   backend    : TkAgg
   ```

   OR

   ```
   backend    : MacOSX
   ```

Some installations seems to work with the former, and some only with the latter. We are still investigating a consistent solution to this problem. It seems to affect Mountain Lion users most.

- End of Document -