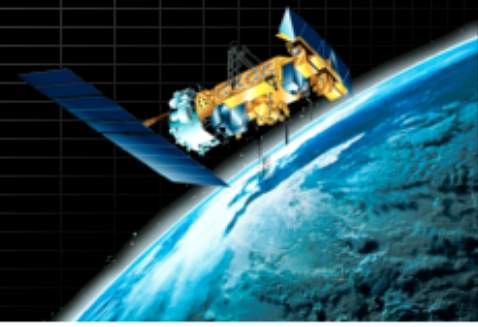


# PPS v2014

## Binary distributions

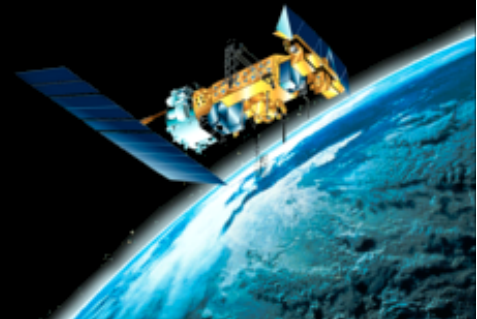


Adam Dybbroe



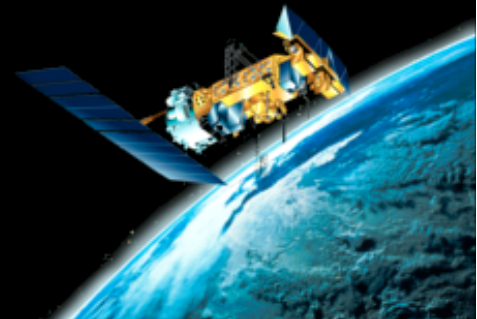
# Outline

- Why binaries?
- Which distributions?
- Content and install destinations?
- What if your OS deviates from those supported?
  - How to build your own binaries



## Why?

- Users have over the years complained that PPS installation is difficult
- This is mainly because the installation of the many PPS dependencies (3<sup>rd</sup> party SW) have been left entirely to the users



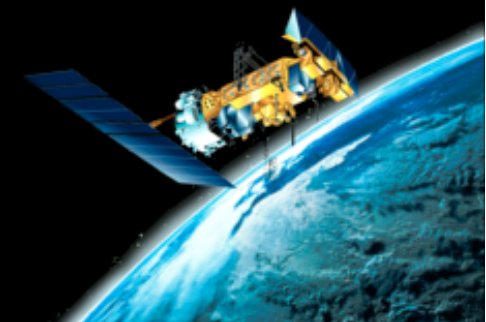
# Under which OS do you intend to install and run PPS v2014?

CentOS 6	
CentOS 7	
RHEL 6	
RHEL 7	
SUSE SLES 11	
SUSE SLES 12	
OpenSUSE 12	
OpenSUSE 13	
Ubuntu Precise (12.04)	
Ubuntu Trusty (14.04)	

Fedora	
Other RPM based	
Other Debian based	
Other Linux	
Non Linux	
Don't know	

## Which distributions

- CentOS-6.4 (RHEL-6.4)
- SUSE SLES 11
- Ubuntu Trusty (14.04 LTS)



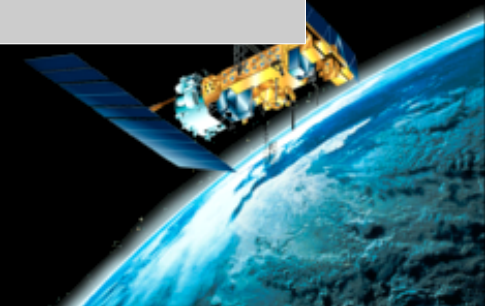
# CentOS 6.4 - 3<sup>rd</sup> party packages

Build by SMHI:

Collected from internet:

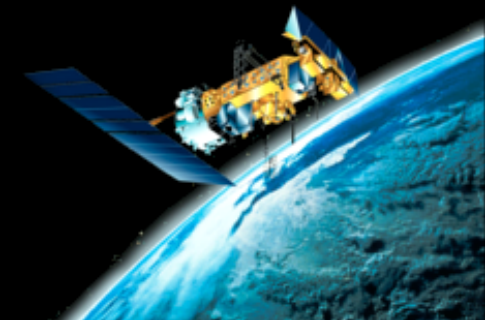
h5py	2.1.2
hlhdf	0.8.1
netCDF4	1.0.9
AAPP	7.6
Proj4 (4 rpms)	4.8.0
pygrib	1.9.6
pyproj	1.9.3
pyresample	0.7.13
RTTOV	11.1
scipy	0.10.1

grib-api (2 rpms)	1.9.18
hdf5 (2 rpms)	1.8.5-patch1
netcdf-4	4.1.1
python-numexpr	1.4.1
python-numpy (2 rpms)	1.6.1



# CentOS 6.4 - PPS

acpg	v2014+20141007-1
acpg-data	v2014+20141007-1
acpg-testdata	v2014+20141007-1
ahamap	v2014+20141007-1
cpp	v2014+20141007-1
pps_nwp	0.4.10-1



## Destinations

- AAPP, ACPG&CPP, AHAMAP, HLHDF, and RTTOV installs under /opt
- PPS data installs under /usr/share/pps
- Other packages installs under /usr

```
[adam@ppscentos ~]$ ls -l /opt/  
total 24  
drwxr-xr-x. 5 root root 4096 8 okt 14.54 aapp  
drwxr-xr-x. 9 root root 4096 8 okt 15.15 acpg  
drwxr-xr-x. 7 root root 4096 8 okt 17.18 ahamap  
drwxr-xr-x. 6 root root 4096 8 okt 17.17 hlhdf  
drwxr-xr-x. 8 root root 4096 8 okt 17.18 rttov-11.1
```



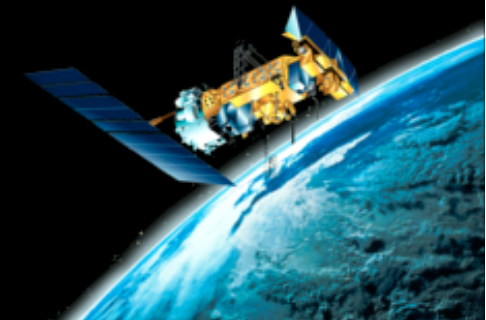


# Destinations

```
[adam@ppscentos ~]$ ls -l /opt/acpg/  
total 84  
drwxr-xr-x. 2 root root 4096 8 okt 17.18 bin  
drwxr-xr-x. 2 root root 4096 8 okt 17.18 cfg  
drwxr-xr-x. 3 root root 53248 8 okt 17.18 cst  
-rw-rw-r--. 1 root root 80 8 okt 13.49 epshdf.pth  
drwxr-xr-x. 2 root root 4096 8 okt 17.18 include  
drwxr-xr-x. 3 root root 4096 8 okt 17.18 lib  
drwxr-xr-x. 2 root root 4096 8 okt 17.18 mkf  
-rw-r--r--. 1 root root 49 8 okt 13.49 RELEASE-VERSION.acpg  
drwxr-xr-x. 2 root root 4096 8 okt 17.18 scr
```

```
[adam@ppscentos ~]$ rpm -qf /opt/acpg/bin/pps_cloudmask  
nwcsaf_acpg-v2014+20141007-1.x86_64
```

```
[adam@ppscentos ~]$ rpm -qf /opt/acpg/bin/cpp-pps  
nwcsaf_cpp-v2014+20141007-1.x86_64
```



# Destinations

```
[adam@ppscentos ~]$ rpm -ql pps_nwp-0.4.10-1.noarch
/usr/cfg/pps_nwp.ini.template
/usr/lib/python2.6/site-packages/pps_nwp-0.4.10-py2.6.egg-info
/usr/lib/python2.6/site-packages/pps_nwp-0.4.10-py2.6.egg-info/PKG-INFO
/usr/lib/python2.6/site-packages/pps_nwp-0.4.10-py2.6.egg-info/SOURCES.txt
/usr/lib/python2.6/site-packages/pps_nwp-0.4.10-py2.6.egg-info/dependency_links.txt
/usr/lib/python2.6/site-packages/pps_nwp-0.4.10-py2.6.egg-info/not-zip-safe
/usr/lib/python2.6/site-packages/pps_nwp-0.4.10-py2.6.egg-info/requires.txt
/usr/lib/python2.6/site-packages/pps_nwp-0.4.10-py2.6.egg-info/top_level.txt
/usr/lib/python2.6/site-packages/pps_nwp/__init__.py
/usr/lib/python2.6/site-packages/pps_nwp/__init__.pyc
/usr/lib/python2.6/site-packages/pps_nwp/__init__.pyo
/usr/lib/python2.6/site-packages/pps_nwp/abparams.py
/usr/lib/python2.6/site-packages/pps_nwp/abparams.pyc
/usr/lib/python2.6/site-packages/pps_nwp/abparams.pyo
/usr/lib/python2.6/site-packages/pps_nwp/config.py
/usr/lib/python2.6/site-packages/pps_nwp/config.pyc
/usr/lib/python2.6/site-packages/pps_nwp/config.pyo
/usr/lib/python2.6/site-packages/pps_nwp/fields/__init__.py
...
```



## Before installation

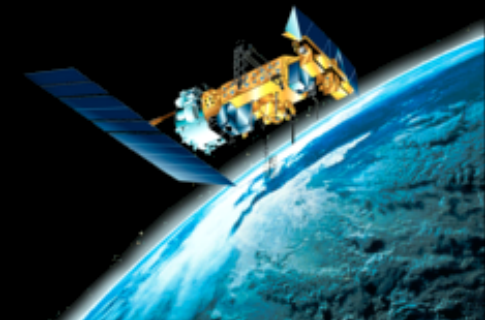
- Make sure the user is allowed to install
- Example – CentOS:

```
[root@ppscentos ~]# less /etc/sudoers
```

```
...
```

```
adam ALL=/usr/bin/yum
```

```
...
```



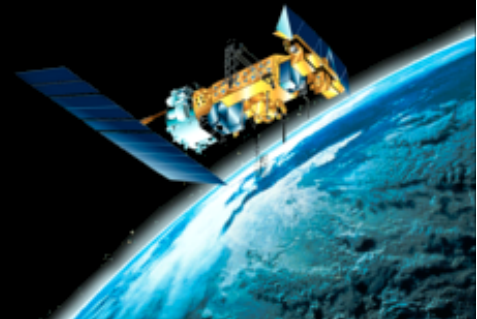
# Installation and verification

- Create 4 directories under  $\${HOME}$ , e.g. SRPM RPM, ppswork, data
- Download packages and data from Help Desk
- Unpack reference data
- Install



# Installation and verification

- Copy the PPS data structure to a writable place
- Unpack the large CPP ancillary data in the right place
- Make a local copy of the .profile\_pps file and change two parameters
- Run the test program(s)
- Run verification script

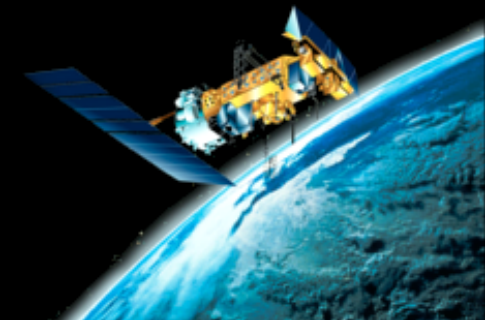


# Installation - Example CentOS:

```
$ cd ${HOME}
$ mkdir ppswork SRPM RPMS data
```

**<Download data package now>**

```
$ cd ${HOME}/ppswork
$ tar xzf ../pps_centos_scripts.tgz
$ cd ${HOME}/data
$ tar xzf pps_refdata.tgz
```

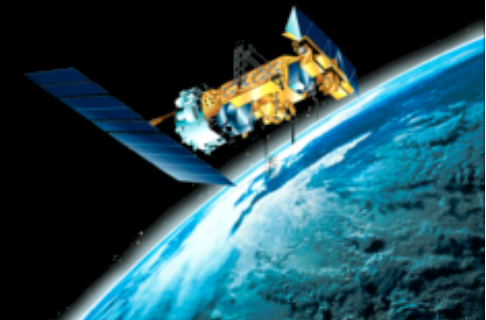


# Installation - Example CentOS:

**<Move rpm files to SRPM and RPMS>**

```
$ cd ${HOME}/RPMS  
$ ../ppswork/centos64_yum_install_pps.sh  
$ cd ${HOME}/data  
$ rsync -a /usr/share/pps/ .
```

**<Now you may uninstall the acpg-data packages>**



# Installation:

```
$ cd import/_CPP_data/source  
$ tar xzf {HOME}/cpp_ancdata_v2014.tgz
```

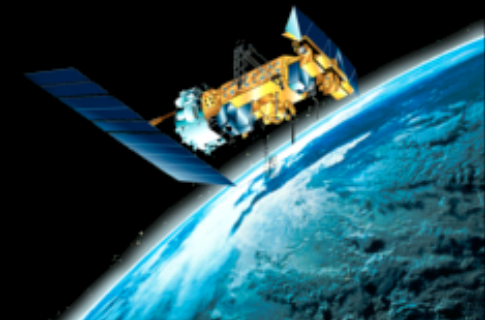
**<Take a cup of coffee...>**





# Installation

```
$ cd ${HOME}/ppswork  
$ cp /opt/acpg/cfg/.profile_pps .
```



# Verification

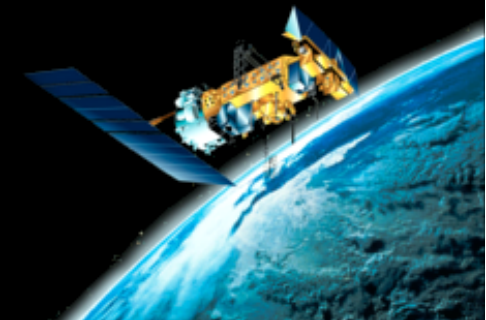
**Edit the file, changing the two lines as below:**

```
DATA_DIR=${HOME}/data ; export DATA_DIR  
PFS_LVL1_DIR=${HOME}/data/import/PPS_data/source ; export PFS_LVL1_DIR
```

**Run the test scripts and verify against reference data:**

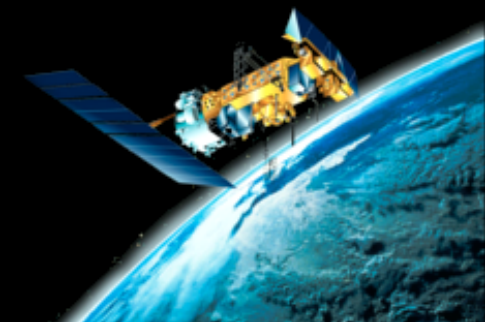
```
$ ./pps_test.sh  
$ ./pps_test_metopgds.sh  
$ python ./pps_check_output.py ${HOME}/data/export  
${HOME}/data/ref_results
```

**Done!**



## Installation and verification

- The verification script checks each output image (png) using ImageMagick and the image metric psnr. If successful (your output is identical or nearly identical to the reference) the script will print an “OK” and a number or “inf” (identical images) for each image compared
- Obviously ImageMagick needs to be installed
- The verification can of course be run on any computer, provided it has access to the data



# When and how to rebuild

- If your OS and version is not among one of the few for which we deliver binary packages, there is a fair chance you can make use of the source packages delivered
- Example - Rebuilding from src.rpm:

```
[adam@ppscentos SRPM]$ rpmbuild --rebuild pyresample-0.7.13-1.src.rpm  
...
```

```
[adam@ppscentos SRPM]$ ls -lt ../rpmbuild/RPMS/noarch/  
total 256  
-rw-rw-r--. 1 adam adam 71948 3 feb 18.14 pyresample-0.7.13-1.noarch.rpm
```

```
[adam@ppscentos SRPM]$ rpmbuild --rebuild nwcsaf_ahamap-v2014+20141007-1.src.rpm  
...
```

```
[adam@ppscentos SRPM]$ ls -lrt ../rpmbuild/RPMS/x86_64/  
total 972  
-rw-rw-r--. 1 adam adam 993213 3 feb 21.55 nwcsaf_ahamap-v2014+20141007-1.x86_64.rpm
```



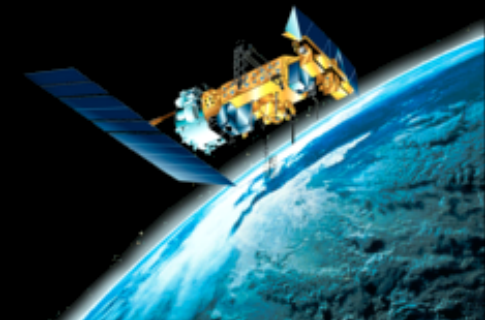
# AAPP is not required

- Only needed in case you want to run PPS on Metop GDS data

```
[adam@ppscentos SRPM]$ rpm -qa | grep nwcsaf  
nwcsaf_ahamap-v2014+20141007-1.x86_64  
nwcsaf_cpp-v2014+20141007-1.x86_64  
nwcsaf_acpg-v2014+20141007-1.x86_64  
nwcsaf_aapp-7_6-1.x86_64
```

```
[adam@ppscentos SRPM]$ sudo yum remove nwcsaf_aapp-7_6-1.x86_64  
Loaded plugins: fastestmirror, refresh-packagekit, security  
Setting up Remove Process  
Resolving Dependencies  
--> Running transaction check  
---> Package nwcsaf_aapp.x86_64 0:7_6-1 will be erased  
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```



...

Remove 1 Package(s)

Installed size: 56 M

Is this ok [y/N]: y

Downloading Packages:

Running rpm\_check\_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Erasing : nwcsaf\_aapp-7\_6-1.x86\_64

1/1

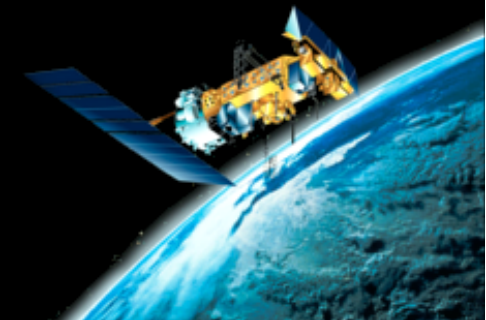
Verifying : nwcsaf\_aapp-7\_6-1.x86\_64

1/1

Removed:

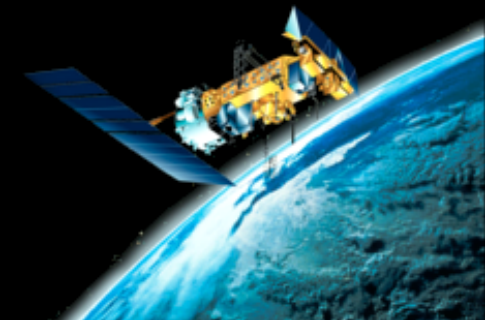
nwcsaf\_aapp.x86\_64 0:7\_6-1

Complete!



## Summary

- In addition to the usual source distribution PPS v2014 is available in binary form for a few Linux distributions
- Installations on other Linux versions than the currently supported ones may be accommodated via the use of source-packages (*rebuild*)







Questions?

<http://www.nwcsaf.org>