

Compile and load ATLOD

ATLOD should be included with all versions of AIPS now

slalib is not needed if you replace libate by a dummy routine:

```

SUBROUTINE EPHMRS (RP_UTCMTAI, RP_C, RP_DJMREFP, RP_DJMREFT,
*      XYUT, NXYUT, CPOS, NCPOS, SU_RA, SU_DEC,
*      EPOCH, PR, PD, APLX, RVEL, DPLX0, NANT, PV, GOAD, PLATE,
*      EPOCHD, RP_PRESSURE, RP_TEMP, RP_HUMID, TLAPSE, TRPAUS,
*      AXIS_OFFSET, PHREF, TAI, INTGRN, RP_DEFEAT, RAP, DAP,
*      TE, DELAY, DEL1D, DEL2D, VEL, B, PARLCT, SW, SW1D)
  write(*,*) 'CALL to libate!!!'
  stop 1
return
end

```

make libate.a from the relocateble. This should never be called for DiFX data.

Add svn properties to source files

In order for your checked in source files to come out with SVN repository version and other useful information, you can do the two following steps:

1. Add stub comment code to yor files, making sure to use legal comment characters. Example for c++:

```

//=====
=
// SVN properties (DO NOT CHANGE)
//
// $Id: $
// $HeadURL: $
// $LastChangedRevision: $
// $Author: $
// $LastChangedDate: $
//
//=====
==

```

2. Set the properties within svn to cause replacement using the following command line

```

svn propset svn:keywords 'Id Revision LastChangedDate LastChangedBy
LastChangedRevision' example.cpp

```

Get IPP version 7 to play nice with mpifxcorr

Please contact Chris Phillips if this is still necessary

Three packages within DiFX depend on IPP: mpifxcorr, vdif_server and difxfilterbank. These instructions apply to all of those.

1. Make a file called `ipp.pc` and install in `${DIFX_PREFIX}/lib/pkgconfig`. The following template should be modified to reflect the path

```
base = /home/swc/difx/intel/ipp/7.0.1.084/ia32

Name: ipp
Description: Intel Performance Primitives
Requires:
Version: 7.0.1.084
Libs: -L${base}/lib/ia32 -L${base}/ipp/lib/ia32 -lipps -lippvm -lippcore -liomp5
Cflags: -I${base}/ipp/include
```

2. Edit the `configure.ac` (or `configure.in`) file for each package, replacing all of the IPP configuration section (ending in `fi`) with the single line

```
PKG_CHECK_MODULES(IPP, ipp)
```

3. Make sure paths to **both** library directories within the IPP install are in the `LD_LIBRARY_PATH`. In the case of the Intel 32 bit version of IPP version 7, this means, for example, replacing:

```
PrependPath LD_LIBRARY_PATH ${IPPROOT}/sharedlib
```

with

```
PrependPath LD_LIBRARY_PATH ${IPPROOT}/lib/ia32
PrependPath LD_LIBRARY_PATH ${IPPROOT}/ipp/lib/ia32
```

in the `setup.bash` file.

4. **Note** that for 64-bit installations, replace `ia32` in the above instructions with `intel64` and you might want to change the version number in the `ipp.pc` file to match more accurately the actual version installed.

Simulate running with infinitely fast datasource

Often it is useful for benchmarking purposes (of either the network or the CPUs) and for debugging certain problems to be able to feed DiFX with a source of fake data. This has been possible for some time with the `difxfake` program (by Chris Phillips). An equivalent feature has been added straight into DiFX version 2.1 and later. This is enabled by changing the `DATA SOURCE` entry for each datastream to `FAKE`. If set, DiFX will ignore the data table and simply flow a perfect stream of senseless data into

the correlator. Everything else about correlation will remain the same: you will still need a .machines file, a .threads file, and all of the other files. The output can be assembled into a FITS file if desired. At the moment no attempt is made to produce test vector data with any particular properties. If there is eventual demand for using this for correctness testing that won't be a difficult task. One could even add to the simulated data bandpass effects, switched power, pulse cal tones, ..., but at the moment these are only possibilities.

From:

<http://www.atnf.csiro.au/vlbi/dokuwiki/> - **ATNF VLBI Wiki**

Permanent link:

http://www.atnf.csiro.au/vlbi/dokuwiki/doku.php/difx/tip_and_tricks

Last update: **2017/03/22 11:58**

