

How to correlate directly off Mark5 modules

This page describes how to set DiFX up to allow “native” Mark5 correlations. This page does not cover the use of fuseMK5. In the “nativemk5” approach described here, an mpifxcorr datastream process runs on each Mark5 unit, meaning that mpifxcorr must be compiled in a manner that can be run on the Mark5 units.

1. Linux requirements

Compatibility of the Conduant StreamStor drivers severely limits the choice of Linux kernels. The supported kernel versions depend on the exact version of the driver (the SDK version number) that is used. SDK version 8.2 or later are suggested as earlier versions have known stability issues related to the way the StreamStor cards are accessed by DiFX. As of this writing (Dec 8, 2011) only 32-bit kernels are supported. See [mixing architectures](#) to see how to configure a cluster to support a mixture of 32 bit and 64 bit members of a cluster running mpifxcorr. Note that mixing different StreamStor SDK versions across the cluster is possible, but for each a separately compiled version of the mpifxcorr binary will be required due to differences in library requirements. Differences between major SDK releases (e.g., SDK8 and SDK9) usually also have API-level changes. Currently mpifxcorr can be compiled against either SDK8 or SDK9 series. There are reports (from RJC: specific failure is on kernel 2.6.26 and SDK 9.0) that Linux Kernel versions greater than 2.6.18 interact badly with the StreamStor/Jungo driver, resulting in garbage data returned for read sizes greater than 8 MB.

2. Installation of Conduant's streamstor Software Development Kit (SDK)

Mark5 units as distributed by Conduant or Haystack Observatory typically have a fully installed software development kit. It may be desirable to reinstall the OS and/or upgrade the SDK. Please follow the guidelines for SDK installation provided by Conduant.

3. Installing a package-config file for streamstor

For mpifxcorr and mk5daemon to find the libraries and include files for the streamstor SDK, it is most convenient to place a package-config (.pc) file in a location identified by environment variable PKG_CONFIG_PATH. Note that the StreamStor libraries must be in the LD_LIBRARY_PATH or in a system library directory for run-time linking. Two examples follow:

Example streamstor.pc for SKD8.2

```
prefix=/usr
exec_prefix=${prefix}
libdir=${exec_prefix}/lib
includedir=${prefix}/include
```

```
Name: streamstor
Description: Conduant streamstor driver and library
Requires:
Version: 8.2
```

```
Libs: -L${libdir} -ldapi921 -lssapi -lrt
Cflags: -I${includedir}
```

Example streamstor.pc for SDK9.0

```
prefix=/usr
exec_prefix=${prefix}
libdir=${exec_prefix}/lib
includedir=${prefix}/include

Name: streamstor
Description: Conduant streamstor driver and library
Requires:
Version: 9.0
Libs: -L${libdir} -ldapi1011 -lssapi -lrt
Cflags: -I${includedir}
```

4. Compilation of "mpifxcorr"

Compilation of `mpifxcorr` should proceed as usual. The configuration process should find the `streamstor.pc` file and properly configure the compilation for native mark5. You should see `nativemk5.cpp` being compiled as a result.

5. Dealing with Mark5 Module directories

Mark5 modules do not contain a conventional file system and due to the potential of failure in decoding modules and the fact that decoding the scans can be time consuming, the reading of module directories to `.dir` files is handled outside the correlation process. These `.dir` files are stored in the directory pointed to by the `$MARK5_DIR_PATH$` environment variable. The program `mk5dir`, found in the `mk5daemon` package, performs the decoding of the modules. This can be run manually on each Mark5 module at the command line:

```
mk5dir A
```

The `.dir` file contains a hash code at the top. If the hash code matches that of the installed module (implying that no new scans have been added and that all existing scans are unchanged) then the directory file is not updated.

Modules containing the new directory format [Mark5 Memo 46](#) can take advantage of a mode where decoding of each scan is not required. The `-fast` (or `-F`) option tells `mk5dir` to take format and timing data straight from the on-module directory with a large decrease in time to decode. This "shortcut" does eliminate a useful sanity check of the module's health that can save time at correlation time.

A "signature" is calculated based on the binary composition of the on-disk header. It is similar to a checksum and is used to determine if the `.dir` file is current with the module. This signature is stored in the 4th column of the first line. It will typically be an 8 to 10 digit number. The signature of an empty directory is 1. This fact can be useful if you want to modify a previously extracted directory to

work on a module that somehow had its on-disk directory erased.

When a module's directory is read, `mk5dir` will, if deemed safe, change the Disk Module State (DMS) to `p`layed. This can be dangerous if there is an SDK mismatch between the unit used to record the module and the one used to play back the module. Unfortunately there is no way to determine the former by looking at the module, though some inferences, of varying degree of certainty, can be made. The setting of DMS to `p`layed can be prevented either by setting the environment variable `DEFAULT_DMS_MASK` to 5 (this is a binary bit mask: 4 = "erased", 2 = "played", 1 = "recorded"), or by specifying one of the `dms` option flags to `mk5dir` on the command line; run `mk5dir -help` for more details on these.

A module directory can be tested for common errors with `checkdir`

```
checkdir NRA0+321
```

Run `checkdir` with `-help` to get additional usage options.

6. Running a correlator job

Within the `.input` file for a correlator job, `nativemk5` is turned on by setting

```
DATA SOURCE:      MODULE
```

And by setting the filename to the Volume Serial Number (VSN) of the module to correlate. Note that only one module VSN can be provided per antenna when using `nativemk5`.

```
D/STREAM 0 FILES:  1
FILE 0/0:          NRA0+293
```

The module to be installed (NRA0+293 in the above example) can be installed in either bank of the Mark5 unit.

Cranky Mark5 modules can be frustrating at a correlator. A module that shows consistent problems can be put into "realtime" mode by changing the `NORMAL` to `RT` at the top of the directory file. This often, but not always, results in better handling of imperfect modules.

7. mk5daemon and extra functionality

From:

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

Permanent link:

<https://www.atnf.csiro.au/vlbi/dokuwiki/doku.php/difx/nativemk5>

Last update: **2015/10/21 10:08**

