VLBA Sensitivity Upgrade Project Memo 29

# The VLBA's Transition to Conduant's SDK9

Walter Brisken

July 20, 2010

**Abstract** Use of Conduant's SDK9 will be required by Mark5C units. This memo illustrates the complexities that must be confronted in migrating from SDK8 to SDK9 on the Mark5 units. Several options for proceeding are presented with comments on impact to VLBA operations. One option has been selected as the most convenient and effective plan. The side effects of this plan almost entirely contained within the VLBA correlator center and will not impact inter-operability with non NRAO facilities.

## 1 Some differences between Mark5A and Mark5C

The main motivation behind the move from Mark5A to Mark5C is the increased bandwdith capabilities. The units differ in several ways:

1. **Data input** Mark5A consumes its data through ribbon cables, passing through the I/O panel, the I/O (Mark5A) board and over the FPDP cable to the Streamstor card. Mark5C receives data in packets over a 10Gb Ethernet port on a daughter board that is attached to the Streamstor card.

2. **Streamstor card** Mark5A units use either the PCI816V100 or the PCI816XF2 variant of the Streamstor card. The Mark5C uses an Amazon Streamstor card with a possible extension to the Amazon Express card in the future.

3. **SDK** The Streamstor cards require FPGA firmware, a operating system driver, and a run-time library for full operation. These three components, and other development-related files, are packed in a "Software Development Kit" (SDK). The Mark5C daughter board requires SDK9 or greater for operation; currently the Mark5A software supports SDK7 and 8 variants. Additional benefits of SDK9 include support for modules larger than 8 TB and eventual support for 64-bit OS.

4. **Recording software** The Mark5A system uses a program called `Mark5A` to control the recording hardware. The Mark5C will use `DRS` (Data Recording System) to do that.

5. **Module directories** The Mark5C system will use a new module directory that is much more capable than the old version. In principle either module directory format can be used by either SDK8 or SDK9, however the various combinations of software/SDK compatibility pretty much dictate that the legacy directory version (now called version 0 in some new VLBA software) is only to be used on Mark5A/SDK8 and the new directory format (called version 1, with a clear path toward future versions) on Mark5C/SDK9. See Mark5 Memo 81 available at http://www.haystack.mit.edu/tech/vlbi/mark5/memo.html.

## 2 Complications

I've spent a good amount of time last week thinking of the cleanest way to support correlation of both Mark5C and Mark5A data on the same correlator (not necessarily cross correlating the two, but simultaneously supporting modules from both types of units). First I enumerate the constraints:

1. The Mark5A units are currently compiled against SDK8. While not out of the question, recompiling Mark5A against SDK9 would be a bit of an effort due to changes in the underlying library API; it is unclear to me at this time what the timescale for such an upgrade would be. Changing Mark5A

to SDK9 would essentially rule out the possibility of us sending modules to other correlators, at least until other correlators are equipped with SDK9 playback units.

2. The Mark5C units require SDK9.
3. Changing between SDK8 and SDK9 is a bit involved and cannot be used to support both types as an operational mode.
4. A module recorded in SDK8 can be correlated by either an SDK8 or an SDK9 playback unit; a module recorded with SDK9 requires an SDK9 playback unit.
5. Automatic erasure of modules at stations requires the correlator to change the disk state from "Recorded" to "Played".
6. Changing the disk state of an SDK8 module in an SDK9 unit will cause the module to be undetected when later inserted in an SDK8 unit.
7. Erasure of any module in an SDK8 unit will allow that module to again be usable by an SDK8 recorder. If the module was recorded in or had its state changed in an SDK9 unit then (potentially error prone) manual reentry of the module VSN is necessary.

This set of constraints is over-determined in the sense that our current module life cycle must be changed to support simultaneous SDK8 and SDK9 operations.

# 3 Solutions

I've come up with a few potential solutions that will allow us to operate in a mixed Mark5 environment:

1. Use a single SDK9 unit at the correlator to read data to a Linux file system and correlate off these files. This would present obstacles to operations and would require a lot of manual effort. Not recommended.
2. Set aside some correlator Mark5 units for the sole purpose of correlating Mark5C data and choose some modules for exclusive use in Mark5C units. At face value this is pretty simple. The complication comes in that all of the correlator code that talks to Mark5 units is shared on a cross mounted partition and the two SDK versions need linking against different libraries. In the end this would entail completely separate installs of DiFX with the associated increase in maintenance needs.
3. Upgrade all of the correlator playback units to SDK9 except one unit which would be used for erasing. The modules would all play back on SDK9. SDK8 modules would become temporarily unusable at an SDK8 recorder until the erasure is performed.
4. Upgrade all of the correlator playback units to SDK9 except one unit which would be used to read directories of SDK8 modules; directory reading of SDK9 modules would proceed on the SDK9 upgraded correlator units. The correlator life cycle would change subtly; instead of the correlator resetting the module state to "Played", the directory reading software would be changed to do so. If module conditioning is required, that can be done for either SDK version on any unit, provided a final reset is performed by a unit of the desired SDK version.

It is the last of these potential solutions (number 4) that I currently favor. This option was tentatively adopted as the path forward on 12 July 2010.

# 4 Implications

Solution number 4 was chosen due to having no implications outside of the VLBA correlator center and minimal implications within. Notably, globals (being correlated either in Socorro our elsewhere) will be unaffected. There are a few noteworthy things to keep in mind:

1. One Mark5 unit, dedicated to SDK8 directory reading, will not be available for correlation.

2. Directory reading of all SDK8 modules must be done with a single unit, which could prolong module check-in a bit, but not by much.

3. The correlator center will be more vulnerable to individual Mark5 unit failures.

4. The `Mark5A` program and its associated utilities will no long be usable on the correlator units (with the exception of the lone Mark5A unit) Much of this functionality is available in a new suite of tools. This suite will have to grow with time to accommodate the needs of operations.

# 5 Updated operations procedures

Assuming solution number 4 above is adopted, the following procedures will be required to minimize the chance of accidental data loss and module confusion. These procedures will simplify once all VLBA Mark5A recorders are converted to Mark5C or returned to Socorro to join the correlator.

To implement some of these procedures it will be very useful to clearly identify on each module whether or not it has SDK9 formatting.

## 5.1 Receiving an SDK8 module

Upon receiving an SDK8 module, its directory should be read using the single remaining SDK8 Mark5 unit. The directory reading can be done either with command line tool `mk5dir`, or by use of the correlator GUI. Reading the directory will cause the module state to change to "Played", causing the auto-erasure once the module is sent back to a station for recording.

If the directory is accidentally read on an SDK9 unit, the module will not be usable by SDK8 until after erasure. It will still be usable by the correlator, but once the module is ready for release it will need to have its VSN reset (using the `vsn` utility) and be erased (using `mk5erase`) on the SDK8 machine.

If the module makes it to a Mark5A unit at a station and has an SDK9 format, it can still be used, but it will have to be manually erased.

## 5.2 Receiving an SDK9 module

Upon receiving an SDK9 module, its directory will need to be read using any of the SDK9 correlator Mark5 units. If put in the SDK8 unit, `mk5dir` will complain that the module is not readable, but no harm will be done.

## 5.3 Correlation

Correlation proceeds in the same way as it does now. While directory reading at correlation time is currently allowed (but discouraged), it will no longer be possible. An error message will be generated if correlation on a module without an up-to-date directory is requested.

### 5.4 Conditioning

Conditioning will be performed using the `mk5erase` program with the addition of the `-c` flag. Conditioning can be performed on any module in any unit. If the target SDK of the recording unit the module is destined for does not match the SDK used in conditioning, the module will need to be put in a unit with the correct SDK and the `vsn` and `mk5erase` (without the `-c` flag) be run to force the correct SDK and directory version.