VLBA Sensitivity Upgrade Memo 30

Enhanced Mark5C Control Specification

Walter Brisken, Matthias Bark, Hichem Ben Frej & Craig Walker

4 Nov. 2010

Abstract Unlike most equipment, the Mark5C data recorders require additional functionality beyond basic configuration. This memo describes those additional requirements. They are largely due to issues related finite capacity of the modules and the non-independence at the signaling level of the two module banks of the recorder. The specifications that follow were initially discussed at the VLBA Software Group meeting on 26 Oct. 2010 and were subsequently refined through email and hallway discussions. Much of the motivation behind the behavior proposed here is based on the relatively successful management of the legacy Mark5A units as implemented within NRAO operations.

1 Bank switching

A new parameter should be added to the `record()` function within the executor that indicates the storage requirements for the scan to be recorded. I suggest this be conveyed in a 64-bit integer representing number of bytes. This is something that would be computed by `vex2script` and should be inclusive of formatting overhead (usually $\sim 1\%$ or less) and is computed only over the data valid period as specified by the vex file. If the storage remaining on the active Mark5 module is less than this quantity *and* a second module with spare capacity is loaded in the inactive Mark5 bank, then one of two actions should happen:

1. the active module should be write protected and the other bank selected for recording, or
2. writing on the active bank should continue until there is no more remaining capacity, at which point this module should be write protected, the active bank should be switched, and recording should resume on the other module.

If no additional module is available, the active bank should continue to record data until its capacity is reached. An alert should be raised as soon as it is clear that insufficient capacity remains.

The decision whether to switch banks immediately (1) or split the scan across two disks (2) should depend on the storage requirements for the scan. For reference remember that at 2 Gbps, 15.4 GB of storage are consumed per minute. The competing goals are minimizing unused capacity and minimizing the number of scans that are split into two chunks. Of all VLBA projects observed in September 2010, the longest scan found was 20 minutes (~ 300 GB) which is $< 4\%$ of an 8 TB module. The experiment in question observed a single source non-stop for hours in 20 minute increments. Aside from that experiment the next longest scan was 10 minutes (~ 150 GB; $< 2\%$). Since modules will only grow larger over time (quite quickly to 16 TB being the norm) a 310 GB limit to the scan size should be used. In the case of insufficient media available in the active bank, a smaller scan would cause an immediate bank switch, anything that size or larger would cause the scan to be split after starting recording on the current active bank. The long term average unused capacity on a module would be considerably less than half the maximum loss due to the large number of short scans that are encountered. The filled (or nearly filled) module should be write protected (with VSI-S command `protect=on`) after the recording is stopped but before switching banks. All of the information required to make this decision is available within the Mark5C MIB, thus this bank-switching functionality should be implemented within the MIB code.

In the case where a long scan is to be interrupted for a module switch, it is advisable to time the end of recording on the initially active module on a convenient time boundary as this will minimize the number of correlation segments required. Rounding down to the 1 minute boundary before the module runs out of capacity would be a reasonable compromise. The start time on the new module should be as soon as

possible. The resumed scan on the initially inactive module should begin as soon as possible and not be constrained to any timing boundary.

The interim Mark5C/RDBE GUI changes the color of a status bar to orange (at 70%) and red (at 90%) as a module fills. Some iteration with operations will be required to determine the most useful indicators of a module nearing full.

In addition to automatic module switching upon exhaustion of capacity, there must be a mechanism for the operators to switch banks.

In the event that the active module is removed, the `DRS` program will switch automatically to the other module, if installed. This module should become active from the point of view of the MIB if allowed.

Upon reboot of the Mark5 unit or restart of the MIB, the module with the largest amount of new data recorded on it (see the DMS discussion under Sec. 3) that is not write-protected should be selected as the active module. If possible, any module selected active by the operator should be remember (by VSN, not bank name) so that upon reboot that module retains its active state, regardless of the bank it was and is in.

2 Module insertions and removals

The Mark5C MIB software should periodically (roughly every 5 seconds) monitor the occupancy of the two banks of the Mark5 unit using the `bank_set?` query. If a change in occupancy is detected through the change of the reported module VSN the MIB should immediately issue an informational alert message indicating to the operator that a module change has occurred. At a convenient time details about a newly inserted module should be collected, including the Disk Module State (DMS), the capacity, details about individual disk drives and the amount of data currently stored on the module. Upon module removal all remembered information about that module should be reset or otherwise marked invalid. The contents of both banks, including information about which bank, if any, is the active bank, should be archived through the monitor data stream after each insertion or removal. and additionally at an interval of about 10 minutes or so to ensure that each 15 minute *chunk* of monitor data contains the information required to know which modules were present over that time range.

2.1 Inactive bank restrictions

Many Mark5C actions cannot be done while recording is in progress. Notably, most access to information about a possible non-active module (i.e., in the other bank) must await a gap in recording. Switching between banks and retrieving information cannot be done instantaneously. Erasure can take a couple seconds to complete. Some observe time will inevitably be lost if recording is requested while these actions are being performed. In small amounts this is fine, but occurrences of these interruptions should be kept minimal. When data is being lost an alert should be raised that would eventually lead to flagging information attached to the astronomy data set. This alert should be lowered once recording has begun. No adjustments to any downstream scheduling (including the record stop time for the current scan) should be made if a delay is incurred in the scan start.

3 Auto-erase

The MIB software should identify insertion of any new module into a Mark5C unit. At the next convenient time (e.g., between requested recordings; see Sec. 2.1) the module's DMS should be determined (possibly requiring temporary bank-switch if another module is active at the time). If the state is `Played`, the module should be erased (using VSI-S command sequence `protect=off;reset=erase`). If the state is `Recorded` or `Erased`, no action is needed. If the state is something else (`Unknown` or `Error`, most likely), an alert should be raised.

The corner case of a new module being inserted as the other, already inserted, module is filling up must be supported as well. In this case, the first module will fill to capacity, stop recording, and at that point the first module shall be erased (if its DMS dictates the need) and recording should proceed.

There should be a mechanism for an operator, through a command line program or GUI, to cause the erasure of a module. Perhaps this should be done through an erase command point in the MIB. Such a commanded erasure should follow the same rules as an auto-erasure and wait for a convenient time to be executed. Any GUI/command line program that allows module erasure should require the explicit entry of the module VSN to prevent accidental erasure; the user interface should do the validity checking and set the erase control point if deemed appropriate. Note that this command point represents a request to erase; actual erasure will wait for a scan gap if recording is in progress.

An alert should be raised for conditions such as module insertion, module removal, module auto-erasure. Such an alert be momentary and informational only and not require any action, possibly other than acknowledgement.

4 Reserved banks

The MIB should implement a logical control point for each bank called `reserved` which, if set, prevents that module from becoming the active bank. This feature is to allow the operator to prevent any new data from being written to a particular bank in cases where the operator needs greater control over the filling pattern. This could be used if one module is to be sent to a different correlator and should therefore be filled only with a single experiment or if a module of a particular capacity is to be saved. An alert should be raised if an automatic bank switch to a reserved module is required. The operator should be able to toggle this control point. Note that the temporary switching of banks for auto-erasure (see Sec. 3) or for obtaining details of a newly inserted module should be ignore the reserved status.

5 Statistics gathering

The Conduant StreamStor card used by all Mark5 units has the capability to produce drive performance statistics that can be useful in quantifying the health of individual drives within a module. These statistics, when enabled, are integrated over the period of a record or playback interval. The basic output values are bin counts for a histogram of transaction durations where an interaction is either the reading or writing of one 65528 byte chunk of data. The last "bin" is the count of transactions that were aborted due to a timeout in reading or writing; when this occurs during data read the data that would have been read get replaced with the fill pattern. Usually any non-zero number in this bin would be considered evidence of a failing disk, or at least one in need of reconditioning. While the time ranges covered by the bins are configurable, the default set of bins that has been the historical standard are as follows:

Bin number	Range (μs)
0	0-1125
1	1125-2250
2	2250-4500
3	4500-9000
4	9000-18,000
5	18,000-36,000
6	36,000-72,000
7	72,000-∞
8	Data replaced

Performance statistics should be gathered for each recorded scan. Prior to each `record=on` command a `start_stats=` command should be issued. Immediately after `record=off` is commanded one `get_stats?` query should be issued for each disk in the module where the number of disk drives can be determined by counting the non-zero entries of the `disk_size?` query. This performance data should be issued as multicast monitor data to be collected and sent to the correlator. Each such record should contain the following information:

1. module VSN (8 characters only),
2. disk number within module (integer between 0 and 7, inclusive),
3. action performed (`Record` if at station, `Play` if at correlator, or `Condition` if the module is being conditioned; the later 2 options would be from Socorro only, except in some debugging or development situations),
4. eight 64-bit integers containing the histogram of read/write times.

Heuristics for a poor performance should lead to alerts being generated for questionable modules. The existing heuristics, implemented in the legacy VLBA interface, can be replicated. Fine tuning based on real performance values will be required for optimal assessment of a module's condition.