Mark 5 Memo #100

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
HAYSTACK OBSERVATORY
WESTFORD, MASSACHUSETTS 01886

Telephone: *781-981-5951*
Fax: *781-981-0590*

27 March, 2014

TO: Distribution
FROM: Chester Ruszczyk, Bob Eldering, Harro Verkouter
SUBJECT: Mark5 User Directory formats

1. User Directories

The Mark5 disk modules have a user definable area of ~ maximum 10 MB of size, in which Streamstor applications can write their own meta data. This memo serves as reference for the layout of the contents as written by the MIT Haystack Mark5A, DIMino, drs programs as well as the jive5ab program by JIVE.

2. Layouts

There are only two different major layouts. The Mark5C layout and the rest. Typically, the actual layout can only be inferred from looking at the actual size of the user directory. See Streamstor API call `XLRGetUserDirLength` to get the current size of the user directory.

The Mark5C layout will not be discussed because MIT Haystack Mark5 Memo #081 ("Specification for enhanced Mark 5 module directory") covers it in detail. The Mark5C layout is recognizable because its size is modulo 128. Unfortunately there are two collisions between two specific Mark5C user directories and two specific other layouts. Section 4, further below, will discuss this.

The other layout(s), for Mark5A and Mark5B systems consist of two parts:

1. a binary dump of a list-of-structs containing the meta data for each recorded scan for a pre-defined maximum number of scans plus some extra parameters, in total the ScanDirectory,
2. a section, having space to contain information about a maximum number of disks ("DiskInfo" hereafter) and, optionally, information about a companion disk module ("BankB" hereafter).

2.1 ScanDirectory size

There are two different ScanDirectories – the "small" and "large" version. The small version can hold 1024 scans maximum whereas the large version can hold 65536 scans.

The size of the ScanDirectory is *solely* dependant on the maximum number of scans that can be held in the user directory: sizeof(ScanDir) = overhead + nscan * sizeof(scan entry)

Putting in the actual values:

- overhead=32 bytes (2x uint32_t , 2x uint64_t and a double)
- sizeof(scan entry)=80 (64 byte scan label + 2x uint64_t)

yields:

sizeof(ScanDir) = 32 + nscan * 80

Thus we arrive at two distinct sizes for the ScanDirectory:

- 81952 bytes for the 1024 scan ScanDirectory
- 5242912 bytes for the 65536 scan ScanDirectory

2.2 DiskInfo/BankB size

All different sizes arise from differences in size of the second part of the user directory – the DiskInfo part and wether or not BankB information was recorded[1].

There are two possible options for the DiskInfo part, one where it can hold information for at most 8 disks and one where it can hold information for up to 16 disks[2]. It is a sad fact that the size of the DiskInfo area is Streamstor SDK version dependant: in SDK9 one of the fields in a structure in the SDK was changed from a 4-byte field to an 8-byte field, so this adds another factor of two in possible sizes.

If the companion module information was recorded in this area as well it adds 64 bytes to the size of this area, creating a third binary option.

The following table summarizes the sizes of the DiskInfo/BankB section of the user directory for all 2^3 possible combinations. Any of these sizes may be appended to any of the two ScanDirectory sizes to produce a total user directory size.

SDK8/ 8DSK	768
SDK8/16DSK	1472
SDK8/ 8DSK/B	832
SDK8/16DSK/B	1536
SDK9/ 8DSK	800
SDK9/16DSK	1536
SDK9/ 8DSK/B	864
SDK9/16DSK/B	1600

The attentive reader will have noticed that there are two distinct combinations yielding the same size: SDK8/16DSK/B and SDK9/16DSK; both will add 1536 bytes to the ScanDirectory. Section 4 below will discuss this.

3 Proposal for normalized naming scheme of user directory layouts

jive5ab has implemented a 'unified' naming scheme to uniquely identify the layouts.

`Mark5CLayout` (obvious)

[1] We should stress that for correlator use, the information contained in this second part of the user directory is mostly irrelevant and has no influence on interpreting the content in the ScanDirectory at all. As such interpretation of it could be skipped altogether.

[2] A binary dump of 8 or 16 instances of `struct S_DRIVEINFO`, as documented in "StreamStor SDK9 User's Guide" documentation, p. 164.

The other layouts follow the following scheme:

```
      Mark5[A|B][8|16]DisksSDK[8|9]{BankB}
```

where values in [..|..] mean alternatives and values in {...} mean optional.

`Mark5A` indicates a 1024 scan ScanDirectory, `Mark5B` indicates a 65536 scan ScanDirectory.

Examples: `Mark5A8DisksSDK8` (the oldest layout), `Mark5B16DisksSDK9` (the latest default) or `Mark5A16DisksSDK8BankB`.

4 Resolving "collisions" in total user directory sizes

Under certain circumstances, there will be different user directories that end up with the same total size. For these user directories, size cannot, obviously, be used to tell the different interpretations apart. Fortunately, the amount of collisions is limited and with the help of some heuristics, code may still be able to infer the correct interpretation. In all cases, the assumption is that when the chosen interpretation is not the actual one, one would be interpreting 'random' bytes as structure fields. This would lead to easily detectable contradictions or impossibilites (or the lack thereof), assuming the current interpretation.

The currently known two collisions are the following:

4.1 `Mark5CLayout` vs. `Mark5A8DisksSDK9BankB` or `Mark5B8DisksSDK9BankB`

The total user directory size for `Mark5A8DisksSDK9BankB` is 82816 bytes, for `Mark5B8DisksSDK9BankB` it is 5243776 bytes. Both of these numbers are integral multiples of 128 bytes thus this collides with the `Mark5CLayout`, which is always modulo 128 bytes.

The Mark5C user directory size increments by 128 bytes for every scan recorded, with an added 128 bytes for the user directory header. As such, these collisions occur only for Mark5C user directories with specific amounts of recorded scans:

- 646 Mark5C scans + 128 bytes header = 82816 bytes
- 40966 Mark5C scans + 128 bytes header = 5243776 bytes

The scan entries in the Mark5A/Mark5B layouts are 80 bytes each, whereas they are 128 bytes in the Mark5C layout. As such, the addresses of the fields in the entries (e.g. the addresses of the start- and end byte fields) will not overlap – at least not for the whole range of scans.

Heuristics that can be used is the knowledge that for either of these two colliding user directories to be a valid `Mark5CLayout`, the `start_byte` field of scan #N must be equal to the `stop_byte` field of scan #N – 1 *for all* 646 or 40966 scans and/or that the `start_byte` number should be incrementing. Another one is that there is a field `scan_number`, which should be equal to N+1, for all of the recorded scans.

4.2 `16DisksSDK8BankB` vs `16DisksSDK9`

Due to a 4-byte difference in one of the disk info structure fields between SDK8 and SDK9, the DiskInfo/BankB section with 16 disks SDK8 + 64 bytes BankB is equal to 16 Disks SDK9 without BankB.

The collision between these two is also only resolvable using heuristics. The application ideally has both the SDK8 and SDK9 definitions internally, interprets the DiskInfo/BankB section of the user directory assuming either of the layouts and starts looking at fields in the structs to decide wether they make sense or not.

The approach jive5ab has implemented first inspects `Capacity` field of the `S_DRIVEINFO`. If it is non-zero, there is some disk information expected (a `Capacity` of 0 implies "no disk at this position"). In this case, it is expected the `Model`, `Serial` and `Revision` character string fields are non-empty.

If the 16DisksSDK9 layout is assumed and it is, in reality, the 16DisksSDK8BankB, then the BankB area will be interpreted as `S_DRIVEINFO` and would hopefully lead to an inconsistency in the assumptions.