

To: Distribution  
From: T. H. Van Vleck  
Date: December 19, 1975  
Subject: Changes to User Ring Programs for New Storage System

### INTRODUCTION

This memorandum describes the necessary changes to the system command and subroutine library and to user programs to handle the new interfaces provided by the new Multics Storage System.

### NEW STATUS CODES

Several new status codes can be returned by the storage system when users attempt operations present in the current system.

#### Logical Volume Not Mounted

The code `error_table_$mount_not_ready` indicates that the storage for a segment resides on a non-permanent volume which the user has not mounted. This code should be returned only for attempts to reference the data or VTOC attributes of a segment; that is, it should be possible to list the ACL of a segment or to rename it even though its logical volume is not mounted. The code can be returned by `initiate`, `truncate`, `delentry`, and by the following other `hcs_` entries:

```
set_max_length, set_max_length_seg  
reclassify, access_class_check  
status_long, status_  
star, star_list, list_dir
```

---

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

Physical Volume Not Found

This code is returned if the physical volume unique ID for a segment cannot be translated into a pvt index. If the logical volume check above has succeeded, this code indicates that the physical volume has been deleted from the logical volume, perhaps due to damage to the volume, or that the branch is damaged, or that the logical volume is being demounted. This code can be returned as a subcondition of seg\_fault\_error.

Segment Is Page Control Out Of Service

The code error\_table\_\$seg\_busted is returned as a subcondition of seg\_fault\_error when an I/O error has occurred on the segment. A new hcs\_entry is provided to reset the switch which causes this error.

Logical Volume Full

The status code error\_table\_\$log\_vol\_full is also a subcondition of seg\_fault\_error. It occurs when a segment cannot be moved to any other pack in a logical volume, and when the pack the segment is on is full.

Physical Volume Out of Service

The status code error\_table\_\$volume\_error results when a physical volume is determined to be out of service. This code can be returned as a subcondition of seg\_fault\_error.

Connection Failure

The status code error\_table\_\$vtoc\_connection\_fail is returned as a subcondition of seg\_fault\_error when a segment's branch does not match its VTOC entry. The condition may arise due to unique ID or directory switch mismatch.

New Case for Access Isolation Error

The code `error_table_$ai_restricted` can be returned for a new case, if the access class of the segment or directory being appended does not fall within the access class bracket of the logical volume where the storage will reside.

Master Directory

The status code `error_table_$master_dir` will be returned when attempts are made to move quota down to or up from a master directory. This error will also occur for attempts to delete an empty master directory from the user ring. (If the directory is nonempty, `error_table_$fulldir` will be returned.)

QUOTA

In the old storage system, quota was a one-dimensional quantity. In the new system, quota on one logical volume is different from quota on another, and those programs which manipulate quota must be aware of the difference. The system will refuse to move quota down to or up from a master directory. In addition, every directory now may have two quotas, one for directory pages and one for segment pages.

Directory quota is handled just like regular quota. New tools commands will be provided as follows:

```
set_dir_quota
move_dir_quota
get_dir_quota
```

The administrative tools will not, initially, set a directory-page quota on project directories; it will therefore be an installation option whether to use directory quota at all. A site which wishes to use directory quota will modify its `master.ec` to give a directory quota to each project directory.

The administrative tools for charging for disk usage will be modified to record the `sons_lvid` for each directory with quota when collecting disk usage information. The `sons_lvid` will be returned in the argument which now contains "infqcnt," the inferior quota count, which is not maintained under the new storage system. The program `charge_disk` will be modified to charge only for page-seconds on system-owned volumes. The system tools for disk usage recording will be modified to be usable by

the owners of private volumes so that they can manage quota on their private packs as well.

### CHANGES TO STORAGE SYSTEM PRIMITIVES

In addition to returning new status codes in the circumstances described above, the storage system primitives have changed in several other respects.

#### Status

The entry `hcs_$status_long` will return the logical volume id (lvid) for a segment in the field now labeled "account." In the current system this field is always returned zero. For directories, the lvid is always the root logical volume: so `status_long` will return the `sons_lvid` for the directory instead.

Both `hcs_$status_` and `hcs_$status_long` will return a single bit for directories in what is now a pad field, to indicate whether a directory is a master directory or not.

All entry points to status including `$status_minf` and `$status_mins` will return `error_table_$mount_not_ready` if the logical volume which contains the contents of the segment has not been mounted by the process. When this status code is returned, any information which is determinable from the directory (which is always online) will be filled in.

The entry `status_for_backup` will return physical volume id as well as lvid, `sons_lvid`, and `master_dir` switch. The number of words returned by this entrypoint may increase.

#### Star

Both `hcs_$star` and `hcs_$star_list` will work whether or not the logical volume which has the storage for the sons of a directory has been mounted by the process; but `error_table_$mount_not_ready` will be returned along with whatever valid information is available. This situation will occur even if a directory has no segments in it (and thus no information is unavailable).

Similar changes will be made to `list_dir`.

Application programs must not use values of the following items about segments when the status code `$mount_not_ready` is returned:

- date and time modified
- date and time used
- current length
- records used

Complete information is returned about directories and links in all cases.

A new entry point for the use of the `list` command will be provided. It is described below in the section on `list`.

### Delentry and Truncate

These primitives work normally unless the logical volume containing the segment has not been mounted by the process. If the volume is not mounted, `error_table_$mount_not_ready` is returned.

This change means that there are some segments which a process may find it cannot delete, and cannot use.

### COMMAND CHANGES

#### list

Calls to `hcs_$star_list` will take much more real time and some additional CPU time because the VTCC reads needed to obtain the dates and records used will, in general, require an arm motion for each entry.

The `list` command will be modified to change the standard output as follows: instead of `records used`, `list` (`bitcount/36864`). Also, `list` date and time branch modified when `-dtm` is requested, only returning date and time segment modified when a new control argument, `-dtcm`, is supplied. A new entrypoint to `star_` will be provided which returns all the information which can be given without going to the VTCC.

For those entries to `star_` which do require VTCC I/O,

realtime wait should be improved by sorting the entries by physical volume id and VTOC index, to attempt to minimize seek time.

### Dir info

The programs save\_dir\_info, comp\_dir\_info, and list\_dir\_info deal with segments which contain listings of a directory's contents. These programs must be updated to handle at least the lvid (sons\_lvid for directories) and master\_dir switch. Physical volume id (pvid) should probably also be saved and checked. These changes will necessitate a redeclaration of the dir\_info segment and compatibility code so that old and new formats can be compared and listed. The lvid and pvid should be stored in character string form rather than in binary.

### New Utility Subroutine

As part of the volume registration package the system must be able to translate from a 36-bit binary lvid to a 32-character logical volume name, and similarly for pvid. The hardware has no knowledge of the logical or physical volume name: this information must be retrieved from the ring-1 RCP data bases. This poses a slight problem: should the user be able to determine the name of volumes to which he doesn't have access?

### Question Handlers

The programs nd\_handler\_ and dl\_handler\_ should be modified to know that some segments cannot be deleted, and to tell the user why.

### Create dir

The create\_dir command will need to accept a new control argument,

-volume LVname

and know, when this argument is specified, that -quota must also be given and that master directory control (mdc\_) must be called.