To: MTB Distribution

From: Larry Johnson

Date: February 16, 1976

Subject: Master Directory Control

INTRODUCTION

The new storage system has introduced the concept of the logical volume (see MTBs 110 and 229). The organization of the hierarchy for the new storage system requires that all the pages of a segment reside on one physical volume. Also, all the segments in a directory must reside on the same logical volume. Each directory, therefore, has a son's logical volume id which specifies the logical volume on which segments in that directory are created. The normal or default case is for a newly created directory to inherit this property from its parent. Quota on such a directory can only be set by moving it from its parent. With NSS, it will be possible to specify the logical volume on which segments are to be placed when the directory is created. When this is done, the directory is known as a master directory. (This is true even if the son's logical volume id is the same as what the default would have been.)

Since the segments in a master directory do not necessarily reside on the same logical volume as segments created it its parent, it follows that the quota in a master directory cannot be charged against its parent. Instead, each logical volume will have associated with it a list of quota accounts against which quota for master directories is charged. Each quota account will have a volume quota assigned, which is the total amount of quota that may be moved to master directories charged against the account. The current quota rules say that if a directory has a zero quota, space used in that directory is charged against its parent. Since this cannot be allowed for a master directory, it follows that a master directory must always have a non-zero quota.

Management of logical volume quota will be the responsiblity of a new ring 1 system called master directory control. Creation and deletion of master directories will be done via ring 1 calls which validate quotas and pass the calls on to ring 0.

Once created, there is very little difference between a master directory and a normal directory.

MASTER DIRECTORY CONTROL USERS

There are several classes of users that must interact with master directory control.

First, is the volume owner. His primary interest is in the area of accounting and quota allocation. Accounting information (in the form of time record products for each user) will be kept and a mechanism will be available for the volume owner to retrieve and reset this information. The volume owner controls quota by designating certain people, by means of the access control segment for the volume, as volume executives.

The volume executive has the capability to set each quota account's volume quota for the volume. He will also have tools available to return information on what quotas have been assigned, master directories created, etc.

Finally, is the volume user. Any user who has a quota account on a logical volume will be able to create master directories on that volume. Other users will still be able to access segments on the volume subject to the normal access rules, and their ability to mount the volume.

MASTER DIRECTORY CONTROL SEGMENT

Each logical volume will have a master directory control segment (MDCS) associated with it. This will be a ring 1 segment kept on the root logical volume, probably in some permanent directory (for example, >sc1>mdc>volume.mdcs).

The MDCS will contain two tables. The first is a per quota account table with the following information:

quota account name
volume quota
volume quota used
time record product of deleted directories
backup accounting data

The quota account name in this table is a two component Personid.Project. Stars are allowed. The account names are therefore ordered as they would be on an acl; names with no stars are first, *.* is last. A user's quota account is the first name in the quota account list that matches the user's group id.

The second table is a per master directory table containing the following:

UID pathname of the directory
quota
owner name
quota account name

The quota account name and the owner name need not, in general, be the same. The owner name is set to the group id of the process creating the directory (although it may later be changed). The quota account name is the name of the quota account against which quota for the directory is drawn. This account name may contain stars. The quota account name must be remembered so that quota adjustments may be made against this account, and so that the quota may be returned when the directory is deleted.

MASTER DIRECTORY OPERATIONS

Creation

A master directory is created thru a ring 1 call which must specify the logical volume and quota, in addition to the parameters required to create a normal directory. All the following tests must be passed before the directory is created.

1. The logical volume must be registered.
2. The user must have a quota account for the logical volume that has sufficient quota to satisfy the request.
3. The user must have append access on the parent directory.

If all these tests are passed, master directory control will call into ring 0 to create the directory, adjust the quota account's volume quota used, and add the new directory to the list in the MDCS. The owner name is set to the name of the process creating the directory.

Quota manipulation

Quota may be changed on a master directory once it is created. The quota is moved back and forth between the volume quota for the quota account in the MDCS and the directory, in a way similiar to quota manipulation on a normal directory and its parent. The following conditions must be met to move quota:

1. The user must be the owner of the master directory, a volume executive, or have the same quota account as the directory.
2. If quota is being added, the quota account must have sufficient quota to satisfy the request. If quota is being subtracted, the directory cannot be left in a record quota overflow condition.
3. The user must have modify access on the master

directory.

If all these conditions are met, the quota can be moved.

Deletion

When a master directory is deleted, its quota will be subtracted from its quota account's volume quota used in the MDCS. Its time record product will also be recorded for accounting by the volume owner. Before the directory can be deleted, the following tests are performed:

1. The user must have modify access on both the directory and its owner.
2. The directory must be empty. (The delete command and delete_ subroutine will delete a master directory's contents as they would for a normal directory, if the logical volume is mounted.)

If these test are passed, the directory is deleted by calling ring 0 and the quota is returned to the MDCS. Note that no special privlidges over and above what is required to delete a normal directory are required to delete a master directory.

Quota Management

Adjusting a quota account's volume quota on a logical volume can be performed only by a volume executive. A ring 1 procedure will be provided to do this.

USER COMMANDS

The user command interface to master directory control is designed to be as similiar to normal directory commands as possible. The following commands will be modified or created:

create_dir	One additional control argument will be added to the existing command. The -logical_volume (-lv) argument specifies the logical volume for which the directory is for, and along with the -quota argument is all that is needed to create a master directory.
delete_dir	The delete_dir command will delete a master directory. There is no change to the command format.
set_mdir_quota	The set_mdir_quota command will set quota on a master directory. Ring 1 will check for sufficient quota in the MDCS to satisfy the request.

list_mdir	The list_mdir command will enable the user to obtain quota information of a logical volume, and list his master directories for that volume.
[logical_volume]	This is an active function which takes a pathname argument and returns the character string name of the logical volume it is on.

VOLUME EXECUTIVE COMMANDS

set_volume_quota	This command will allow the volume execute to set or adjust any user's volume quota for a logical volume.
list_mdir	This is the same command used by normal users to list master directories and quotas. Additional control arguments can be used by volume executives to specify what users should be included.
set_mdir_account	This command is used to set the quota account name for a master directory. Quota for the directory would be returned to the old account name and subtracted from the new account name.
set_mdir_owner	This command is used to change the owner name for a master directory.

VOLUME OWNER COMMANDS

The volume owner controls executive access to the logical volume through the standard acl mechanism. he will also require some method of obtaining accounting information.

Name: create_dir, cd

The create_dir command causes a specified storage system directory branch to be created in a specified directory (or in the working directory). That is, it creates a storage system entry for an empty subdirectory. See the description of the create command for information on the creation of segments.

Usage

```
     create_dir paths -control_args-
```

where:

1.	paths	specify the names of the subdirectories to be created.
2.	control_args	may be chosen from the following:
	-access_class XX, -acc XX	applies to each path*i* and causes each directory created to be upgraded to the specified access class. The access class may be specified with either long or short names.
	-logical_volume V, -lv V	specifies that each directory created is to be a master directory whose segments are to reside on logical volume V.
	-quota *n*	causes a quota of *n* to be moved to the created directory. A quota should be specified when creating a master directory and/or an upgraded directory (see "Notes" below).

Notes

The user must have append permission on the containing directory of each directory created.

If the creation of a new subdirectory would introduce a duplication of names within the directory, and if the old subdirectory has only one name, the operation is not performed. If the old subdirectory has multiple names, the conflicting name is removed and a message to that effect issued to the user.

The user is given sma access on the created subdirectory.

All superior directories specified in path*i* must already exist. That is, only a single level of storage system directory hierarchy can be created in a single invocation of the create_dir command.

In order to create a master directory, the user must have a quota account on the logical volume with sufficient volume quota to create the directory. A master directory must always have a non-zero quota; therefore the -quota argument must always be given. A master directory can be created even though the logical volume is not mounted.

Each upgraded directory must have a quota greater than zero and must have an access class that is greater than its containing directory. The specified access class must also be less than or equal to the maximum access authorization of the process. If a quota is not specified for an upgraded directory, a quota of 1 will be assumed. (See Section III in the MPM Subsystem Writers' Guide for further details.)

When the -access_class control argument is specified, the command does not create a new directory through a link. Creating through links is allowed only when the access class of the containing directory is taken as the default.

Examples

```
     create_dir sub >my_dir>alpha>new
```

creates the subdirectory sub immediately inferior to the current working directory and the subdirectory new immediately inferior to the directory >my_dir>alpha. As noted above, the directories my_dir and alpha must already exist. Both directories are assigned the access class of their containing directory.

```
     create_dir subA -access_class a,c1,c2 -quota 5
```

creates the subdirectory subA with an access class of a,c1,c2 and a quota of 5 pages. The directory subA would be created immediately inferior to the working directory. (The access class names a, c1, and c2 used in the example represent possible names defined for the site. See the print_auth_names command for more details on access class names.)

```
     create_dir subB -logical_volume work -quota 100
```

creates a master directory subB in the working directory. Segments created in this new directory will reside on the logical volume work. The directory subB is given a quota of 100 records.

Name: delete_dir, dd

The delete_dir command causes the specified directories (and any segments, links, and multisegment files they contain) to be deleted. All inferior directories and their contents are also deleted. See the descriptions of the delete and delete_force commands for an explanation of deleting segments and deleting protected segments, respectively.

Usage

```
     delete_dir paths
```

where paths are the pathnames of the directories to be deleted.

Notes

The user must have modify permission for both the directory and its superior directory. The star convention can be used. Before deleting each specified directory, delete_dir asks the user if he wants to delete that directory. It is deleted only if the user types "yes".

If deleting a non-empty master directory, or a directory containing inferior non-empty master directories, the user must have previously mounted the logical volume(s). If a non-empty master directory for an unmounted volume is encountered, no subtrees of that master directory will be deleted, even if they are mounted.

Warning: Protected segments in path*i* or any of its subdirectories are deleted. Segments whose write bracket is less than the current ring are not deleted; consequently, the subtree being handled is not completely deleted if any such segments exist in the directory. For a discussion of ring brackets, see "Intraprocess Access Control (Rings)" in Section III of the MPM Subsystem Writers' Guide.

Name: move_quota, mq

The move_quota command allows a user to move records of quota between two directories, one immediately inferior to (contained in) the other.

Usage

```
     move_quota path1 quota_change1 ... pathn quota_changen
```

where:

1.	pathi	is the pathname of a directory. The quota change takes place between this branch and its containing directory. A pathi of -wd or -wdir specifies the working directory. The star convention cannot be used.
2.	quota_changei	is the number of records to be moved between the superior (containing) directory quota and the pathi quota. The quota_change argument can be either a positive or negative number. If it is positive, the quota is moved from the containing directory to pathi; if it is negative, the move is from pathi to the containing directory.

Notes

The user must have modify permission on both the directory specified by pathi and its containing directory.

After the change, both directories (pathi and its parent) must have a quota greater than or equal to the number of records used in each directory, unless the change makes the quota on pathi zero.

If the change makes the quota on path*i* zero, there must be no immediately inferior directory with nonzero quota, and the records used and the record-time product for path*i* are reflected up to the superior directory.

If path*i* is an upgraded directory (its access class is greater than the access class of its containing directory), quota_change*i* must be positive. Quota can only be moved back to the containing directory of an upgraded directory by deleting the upgraded directory.

Quota may not be moved between a master directory and its parent. See the set_mdir_quota command for changing the quota on a master directory. Quota may, however, be moved between a master directory and an inferior directory as described above.

Example

```
move_quota >udd>m>Smith>dir1 1000 >udd>m>Smith>dir1>dir2 -50
```

adds 1000 records to the quota on >udd>m>Smith>dir1 and subtracts 1000 records from the quota on >udd>m>Smith. It then subtracts 50 records from the quota on >udd>m>Smith>dir1>dir2 and adds 50 records to the quota on >udd>m>Smith>dir1.

Name: set_volume_quota, svq

The set_volume_quota is used to set a quota account's volume quota on a logical volume. This command is to be used by volume executives.

Usage

```
     set_volume_quota volume account change
```

where:

1.	volume	is the name of the logical volume for which quota is to be set.
2.	account	is the name of the quota account the volume quota is for.
3.	change	is the amount of quota, or the amount of quota change, and can be specified as follows:
	+n	n records will be added to the quota
	-n	n records will be subtracted from the quota
	n	the quota will be set to n

Notes

To use this command, the process must have "e" access to the logical volume. It is not necessary that the volume be mounted.

If the volume quota is set less than the quota account's current quota used, the quota is changed as directed, but a warning message is printed.

Name: list_mdir, lmd

The list_mdir command allows the user to print information on logical volume quotas and master directories.

Usage

```
     list_mdir volume -control_args-
```

where:

1. volume — is the name of the logical volume

2. control_args — may be selected from the following:

-quota	specifies that only quota information is to be printed.
-directory, -dr	specifies that only master directory information is to be printed.
-brief, -bf	suppresses headings and shortens the output lines (see examples).
-long	cause more complete information to be printed, including the quota account for each directory.
-owner list	specifies a list of directory owners for which information is desired. The default is to print only information for directories owned by the user issuing the command. The star convention may be used for owner names.
-account list	specifies a list of quota account names for which information is desired. The star convention may not be used. Stars in account names will only match quota account names which contain stars.
-all	specifies that information is required for all users of the logical volume.

Notes

If neither the -quota or -directory arguments are specified, information about both quotas and directories will be printed.

It is not necessary that the logical volume be mounted to use this command.

Executive access to the logical volume is required to use the -owner, -account, and -all control arguments. If -all is specified, -owner and -account must be omitted. If both -owner and -account are specified, information will only be printed for directories which match both conditions.

Examples

```
1. ->     lmd fred            (issued by user George)
          QUOTA    PATHNAME
          100      >udd>m>George>sub
          250      >udd>m>George>sub1>sub2
          350 records of quota assigned, 500 available.

2. ->     lmd fred -bf        (issued by user George)
          >udd>m>George>sub
          >udd>m>George>sub1>sub2
          Quota=500, used=350.

3. ->     lmd fred -quota -bf (issued by user George)
          Quota=500, used=350.

4. ->     lmd fred -all       (issued by a volume executive)
          OWNER               QUOTA    PATHNAME
          George.Multics      100      >udd>m>George>sub
          George.Multics      250      >udd>m>George>sub1>sub2
          Ralph.Multics       900      >udd>m>Ralph>work

          ACCOUNT             USED     VOLUME QUOTA
          Ralph.Multics       900      1000
          *.Multics           350      500

          Total volume quota: 1500
          Total quota used: 1250
```

Name: set_mdir_quota, smdq

The set_mdir_quota command is used to set the quota on a master directory.

Usage

```
     set_mdir_quota path1 change1 ... pathn changen
```

where:

1.	pathi	is the pathname of a master direcotry whose quota is to be changed.
2.	changei	is the amount of quota or quota change and can be specified as follows:
	+n	add n records of quota to pathi
	-n	subtract n records of quota from pathi
	n	set the quota on pathi to n

Notes

The user must have modify access on the directory, and must meet one of the following conditions:

1. Be the owner of the master directory, or
2. Be a volume executive, or
3. Have the same quota account as the master directory

If the quota is being increased, the master directory's quota account must have sufficient volume quota to satisfy the request.

The quota of a master directory can never be zero, and it can never be set less than the current numbers of records being charged against the directory.

<u>Name</u>: set_mdir_owner, smdo

The set_mdir_owner command is used by a volume executive to set the owner of a master directory.

Usage

```
     set_mdir_owner path owner
```

where:

1.	path	is the path name of the master directory to be changed.
2.	owner	is the name of the new owner of the master directory, specified as Personid.Project.

Notes

The user must have "e" access on the logical volume that the master directory is for. The volume need not be mounted.

Name: set_mdir_account, smda

The set_mdir_account command is used by a volume executive to set the quota account of a master directory.

Usage

```
set_mdir_account path account
```

where:

1.	path	is the path name of the master directory to be changed.
2.	account	is the name of the new quota account of the master directory, specified as Personid.Project.

Notes

The user must have "e" access on the logical volume that the master directory is for. The volume need not be mounted.

The quota for the master directory is returned to the old quota account and is withdrawn from the new quota account. The new quota account must have sufficient quota to do this.

Name: mdc_

The mdc_ subroutine (actually a ring1 gate) provides a series of entry points for manipulation of master directories.

Entry: mdc_$create_dir

This entry is used to create a new master directory. Its arguments are roughly analogous to hcs_$append_branchx.

```
     dcl mdc_$create_dir entry (char(*), char(*), char(*),
          fixed bin(5), (3) fixed bin(3), char(*), fixed bin,
          fixed bin(35));

     call mdc_$create_dir (dir, ename, volume, mode, rings,
          user_id, quota, code);
```

where:

1.	dir	is the pathname of the containing directory. (Input)
2.	ename	is the entry name of the subdirectory. (Input)
3.	volume	is the name of the logical volume which is to contain segments created in the new directory. (Input)
4.	mode	is the user's access mode. (Input)
5.	rings	are the ring brackets of the directory. (Input)
6.	user_id	is an access control name. (Input)
7.	quota	is the quota to be placed on the new directory. (Input)
8.	code	is a standard status code. (Output)

Entry: mdc_$create_dirx

This entry is an extension of mdc_$create_dir which is similiar to hcs_$create_branch_.

```
     dcl mdc_$create_dirx entry (char(*), char(*), char(*), ptr,
          fixed bin(35));

     call mdc_$create_dirx (dir, ename, volume, info_ptr, code);
```

where:

4.	info_ptr	is a pointer to a status structure as described under hcs_$create_branch_. (Input)

Entry: mdc_$delete_dir

The entry point is used to delete a master directory.

```
     dcl mdc_$delete_dir entry (char(*), char(*), fixed bin(35));

     call mdc_$delete_dir (dir, ename, code);
```

Entry: mdc_$set_mdir_quota

This entry is used to set the quota on a master directory.

```
     dcl mdc_$set_mdir_quota entry (char(*), char(*),
          bit(1) aligned, fixed bin, fixed bin(35));

     call mdc_$set_mdir_quota (dir, ename, sw, quota, code);
```

where:

3.	sw	is a switch indicating the kind of quota change. "0"b means that the directory quota should be set to the quota parameter. "1"b means that the quota parameter should be algebraically added to the current directory quota. (Input)

Entry: mdc_$set_volume_quota

This entry is used to set the volume quota for a quota account on a logical volume.

```
     dcl mdc_$set_volume_quota entry (char(*), char(*),
          bit(1) aligned, fixed bin, fixed bin(35));
```

```
call mdc_$set_volume_quota (volume, account, sw, quota,
     code);
```

where:

2. account — is the name of the quota account in the form Person.Project. The quota account name may contain stars. (Input)

Entry: mdc_$set_owner

This entry is used to set the owner name of a master directory.

```
dcl mdc_$set_owner entry (char(*), char(*), char(*),
     fixed bin(35));

call mdc_$set_owner (dir, ename, owner, code);
```

where:

3. owner — is the new owner name of the master directory. (Input)

Entry: mdc_$set_account

This entry is used to set the quota account of a master direcotry.

```
dcl mdc_$set_account entry (char(*), char(*), char(*),
     fixed bin(35));

call mdc_$set_account (dir, ename, account, code);
```

where:

3. account — is the name of the new quota account. The directory quota will be returned to the old account and re-drawn from this new account. (Input)