

To: MTB Distribution
 From: T. Casey
 Date: 15 December 1975
 Subject: List Command Changes for New Storage System

This MTB describes, and explains the reasons for, the changes that will have to be made to the list command before the new Storage System is installed. It is written in the form of a notice to users of a change that has already been made.

Introduction

Several of the items printed by the list command are kept in the Volume Table of Contents (VTOC) rather than in the directory, in the new Storage System. They are: date_time_modified (dtm), date_time_used (dtu), and records used.

The VTOC entry (VTOCE) for each segment (or directory) in a directory is accessed individually, via a disk I/O operation. (Directory entries, on the other hand, are held in the pages of a directory, and one page fault could bring into core all of the entries in a small directory.) It is therefore more costly and time-consuming (in terms of real elapsed time) to access both the directory entries and the VTOCEs of all entries in a directory, than it is to access only the directory entries.

Timing tests on the NSS system on the development machine, executing the command "list -total" in the directory >ldd>include (830 entries) give the following results (all figures are times, in seconds):

	Elapsed	Real CPU	Virtual CPU
no VTOC I/O	1.5	1.2	0.7
VTOC I/O	18.0	7.0	5.8

Therefore, the list command has been modified so that, by default, it will print only those items that are kept in the directory. It does, however, have the ability to list items from the VTOC, when the user so specifies. The changes include the ability to print (and sort on) date_time_entry_modified (dtem) as an alternative to dtm, and length (in records) computed from the bitcount as an alternative to records used.

The next section discusses the relationships between those date and length items whose treatment by the list command has been

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

changed, and it shows why the new items are acceptable (and even preferable in some cases) substitutes for the old ones. The final section describes the interface changes to the list command.

Dates and Lengths

The bitcount indicates the position of the last bit of meaningful information in a segment. (It has a different meaning for directories. That will be discussed later.) The last bit falls somewhere within the last page of the segment. The number of the last page (counting from 1) is the current length of the segment. The current length can be computed from the bitcount by dividing the latter by $(36*1024)$ and adding 1 if there is a remainder. The current length is maintained automatically by the storage system. (It is set to the number of the last page containing any non-zero bits.) The bitcount is maintained by the command system. It is set by any user-ring procedure that changes the position of the last meaningful bit in the segment. (The `adjust_bit_count` and `set_bit_count` commands allow the user to set it correctly, or to any arbitrary value, respectively.) Most installed commands set the bitcount of any segment that they modify. Users should make certain that all privately-maintained commands and subroutines do the same.

It should be noted that the above relationship between bitcount and current length does not hold if: 1) the bitcount is set incorrectly; or 2) the last page of meaningful information is all zeros (a very rare case).

The records used of a segment will equal the current length, except when one or more of the pages of the segment contain all zeros (a rare case). Pages of zeros are not stored on disk, or charged to quota, or counted in the records used figure. The list and status commands both print records used. (The status command also prints current length, but only when it is different from records used.) The records used figure is maintained automatically by the storage system.

Whenever the directory entry for an object in the storage system is modified, the `dtem` is updated to the current `date_time`. Setting the bitcount involves modification of the directory entry, and therefore forces `dtem` to be updated.

The `utm` and `dtu` figures are set at the time a segment is deactivated, and they are updated to the current time whenever they are accessed while a segment remains active. On a very lightly loaded system, a segment can remain active for hours, or even until shutdown, causing the `dtem` and `dtu` figures to have a much later value than the user would expect.

Thus, `dtem` can be more accurate than `dtem` as an indicator of the time of last modification. (It must be kept in mind, however,

that `dtm` is also updated by changes to other user-settable attributes, such as: `acl`, `names`, `ring brackets`, `safety switch`, `copy switch`, `max length`, `entry bound`, etc.)

By default, the `list` command will print names, mode, and length computed from `bitcount`. This involves no VTOC I/O. Also, the `date_time_modified` (`dtm`) argument will now cause `dtm` to be used (for sorting or printing), while a new argument causes the `dtm` from the VTOC to be used (see next section). For those segments whose `bitcounts` are (correctly) set whenever they are modified, the output produced by the `list` command using computed length and `dtm` will be identical to that produced using records used and `dtm`, except that the time of modification will be exact, rather than being a function of how quickly the modified segment gets deactivated.

A zero in the length column will usually indicate a segment whose `bitcount` has not been correctly set. Since many commands fail when operating on a segment whose `bitcount` is (incorrectly) set to zero, it is useful to be able to detect such cases by a glance at `list` command output.

When arguments that force VTOC I/O are given, accessing of the VTOCEs is postponed as long as possible. Entries are eliminated on the basis of entry type, starnames, and the `-first N` argument (except when sorting by a VTOC item is also specified), before the VTOCEs of the selected entries are accessed. For this reason, use of `-first N` is recommended over `QUITting`, as a means of listing only the first few entries in a large directory.

Since it is possible to specify sorting on and printing of items separately, it is now possible to specify these two functions inconsistently. Doing so will result in an error message and no other output. Consistency restrictions are given in the control argument and sort key descriptions in the next section. These restrictions can be summarized as follows: the `list` command can operate on only one of the two size figures, and on only one of the two modification dates. Any combination of arguments that specifies sorting on one item while printing the other item, within either pair, is inconsistent and will be rejected.

List Command Interface Changes

Three new control arguments have been added, and one has been eliminated. The new ones are:

`-length`, `-ln` print current length computed from `bitcount`. This argument is inconsistent with `-record`; only one of the two may be given. This argument is the less expensive of the two, and is the default.

`-date_time_entry_modified`
`-dtem` print the `date_time_entry_modified`. This argument is inconsistent with `-date_time_contents_modified`; only one of the two may be given. This argument is the less expensive of the two.

`-date_time_contents_modified`
`-dtdcm` print the `date_time_contents_modified` (formerly called `date_time_modified` or `dtm`). This argument is inconsistent with `-date_time_entry_modified`; only one of the two may be given. This argument is the more expensive of the two.

The `-date_time_modified` (`-dtm`) argument is now obsolete, but it will be accepted as a synonym for `-dtem`. The reason for this is to save users from making unintentional and expensive VTOC references, by typing the more commonly-used of the two date arguments by force of habit.

The `-record control` argument description is changed to read:

`-record, -rec` print the records used. This argument is inconsistent with `-length`; only one of the two may be given. This argument is the more expensive of the two.

Three new sort keys have been added, and one has been eliminated. The new ones are:

`length, ln` sort by length computed from `bitcount`. This argument is inconsistent with the `-record control` argument.

`date_time_entry_modified`
`dtem` sort by `date_time_entry_modified`. This argument is inconsistent with the `-dtdcm` argument. If the `-dtem` argument is given, and no sort key control argument follows `-sort`, then this argument is implied as the default sort key.

`date_time_contents_modified`
`dtdcm` sort by `date_time_contents_modified`. This argument is inconsistent with the `-dtem` control argument. If `-dtdcm` is given and no sort key argument follows `-sort`, then this argument is implied as the default sort key.

The `date_time_modified` (`dtm`) sort key is obsolete, but it will be accepted as a synonym for `dtem`.

The description of the record sort key is changed to read:

record, rec sort by records used. This argument is inconsistent with the -length control argument. If this argument is given, and the default columns have not been overridden by the -brief or -name control arguments, then the -record argument is implied by this argument.

Since the length is computed from the bitcount, and directory bitcounts are never meaningful as length indicators (when nonzero, they are interpreted as msf component counts), only the mode and names columns are printed for directories, by default. The records column is printed for directories only when the -record control argument is given.

Since the -record control argument must now be used to specify that records used from the VTOC, rather than length computed from bitcount, is to be printed, it will no longer have the effect of overriding the column defaults. For consistency, the -mode argument will also no longer have the effect of overriding the column defaults. Now, only the -brief and -name control arguments will have that effect.