To: Distribution

From: Joan Scott

Date: June 8, 1976

Subject: Approved MCR's from May 16 - 30, 1976

Attached are the MCR's which were approved from May 16 - 30, 1976

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.
<table>
<thead>
<tr>
<th><strong>MULTICS CHANGE REQUEST</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TITLE:</strong> Multics Data Base Manager</td>
</tr>
<tr>
<td><strong>AUTHOR:</strong> O. Friesen</td>
</tr>
<tr>
<td><strong>Planned for System:</strong> MR 4.0</td>
</tr>
<tr>
<td><strong>Fixes Bug Number(s):</strong> not applicable</td>
</tr>
<tr>
<td><strong>Documented in MTH:</strong> 271, 272, 273</td>
</tr>
<tr>
<td><strong>Incompatible Change:</strong> no</td>
</tr>
<tr>
<td><strong>User/Operations-visible Interface Change:</strong> no</td>
</tr>
<tr>
<td><strong>Coded in:</strong> (O)PL/I (<strong>)ALM (</strong>)other-see below</td>
</tr>
<tr>
<td><strong>Performance:</strong> ( )better (**)same ( **)worse</td>
</tr>
<tr>
<td><strong>DOCUMENTATION CHANGES (specify Q07.QR MORE)</strong></td>
</tr>
<tr>
<td>MPM (vol/sect) 4/1I:3/III MPAM (sect)</td>
</tr>
<tr>
<td>MOSN (sect) MSAM (sect)</td>
</tr>
<tr>
<td>PLMs (ANM)</td>
</tr>
<tr>
<td>Info Segs</td>
</tr>
<tr>
<td>Other</td>
</tr>
<tr>
<td><strong>MDAM User Supplement (initially all doc. will appear in 4.0 SRM)</strong></td>
</tr>
<tr>
<td><strong>OBJECTIONS/COMMENTS:</strong></td>
</tr>
</tbody>
</table>

**Headings are:** SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

**SUMMARY:** Install the initial release of the Multics Data Base Manager.

**REASON:** To provide Multics with a data base management system.

**IMPLICATIONS:** None.
SUMMARY: In certain circumstances of directory rebuild, a long salvage can cause directory compaction without removing the deposited addresses from the compacted directory's VTOC entry. This causes an unprotected address.

Fix the bug.

REASONS: Unprotected addresses cause reused ones, and reused addresses cause shared data and crashes.
SUMMARY: Unlock the AST during the VTUC entry read during segment activation.

REASONS: Improve performance by reducing AST lock contention.

IMPLICATIONS: Slightly more complicated logic in activate needed to revalidate derived AST pointers. Slightly more process time expended in multiple lock calls.
SUMMARY: Create phcs_$get_vtoc, a privileged gate for obtaining the contents of a VTOC entry in the outer ring.

REASONS: System maintenance and debugging has often found it critical to inspect the contents of VTOC entries, which can only be done now by shutting down, chancy techniques involving VTOC buffer dumping, or risky techniques involving highly privileged access.

IMPLICATIONS: There are many covert and overt information paths through this dumping, but access to phcs_ is limited to trusted system programmers, who could dump passwords out of teletype buffers, etc., with phcs_ as it stands.
**SUMMARY:** Fixes minor bugs and new compiler warning messages
Adds capability of running Tektronix 4002 and 4002A devices

**REASONS:** Better service

**IMPLICATIONS:**
Although the code for the 4002 should reasonably work, I cannot guarantee irrevocably that it does. The model is no longer available from the manufacturer, and a demo model cannot be obtained. However, this was (and still is) a very popular terminal, and some of our sites (MIT, System M for two) do have user-owned 4002's. Users using these devices currently either use the existing 4012 facility, which is not totally satisfactory; of those at MIT use MIT's own 4002 facility, which has its own problems (e.g. text not aligned correctly). Users in the first group will not lose anything by trying the new 4002 package. MIT might be warned that they should keep their facility around just in case for a short time. However, note that non-exhaustive user testing at both sites have indicated that the new 4002 package appears immediately superior to the 4012 package and MIT's 4002 package. Note that both the new 4002 package and MIT's can co-exist at the same time on the same system until they are satisfied that it indeed works.

**DETAILED PROPOSAL:**
Fix bug in graphic editor causing OOB's in new areas.
Cosmetic internal code changes to graphic dim to suppress new compiler warning messages for 'char_string = ""' and 'bit_string = ""' typos that were never previously diagnosed.
Modify graphic_dim to suppress any MCS echo-type modes while performing graphic input, so that random input from graphic devices does not mess up the user's screen.
Add support for Tektronix 4002 and 4002A terminals.
Delete nonstandard obsolete name "gr_print" from bound seg.

OVER
If any questions arise, please CALL 202-695-2478 and I will try to clarify them. Since I am not in Phoenix, it takes too very long to shuffle the forms back and forth, especially when I then have to call anyway to find out exactly what the objections were! My last MCR was submitted 12/03, and took until 04/20 to occur, although it was accepted on 01/19, because it was lost in the shuffle.

Also, documentation changed for this MCR was submitted two months ago or better, in anticipation of a slow publication.
Fix attempt to use null pointer in syserr_log_copy$auto_copy_inita that occurs when the paged syserr log in ring 0 does not exist.

This bug causes a fault in ring 0 during log copying initialization whenever the PART LOG card is missing from the config deck.

This case will be handled properly.

The internal subroutine GET_SLOG_PTR in syserr_log_copy returns a null pointer and a non-zero error code if the paged syserr log does not exist. (This is the case if the PART LOG card is missing.) Change the code in syserr_log_copy$audit_copy_inita to not reference through the pointer returned by GET_SLOG_PTR if the returned code is non-zero.
**TITLE:** Install quick edit-directed pl2 output  
**AUTHOR:** R. Schoeman

|----------------------|-------------------|------------------|------------------|-----|-----|----------|----------|---------|------------------|---------|--------------|

**DOCUMENTATION CHANGES**

<table>
<thead>
<tr>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM (Vol, Sect.)</td>
<td>PLMS (AN #)</td>
</tr>
<tr>
<td>MPAM (Sect.)</td>
<td>MGSM (Sect.)</td>
</tr>
<tr>
<td>MSAM (Sect.)</td>
<td>Info Segs</td>
</tr>
<tr>
<td>Other (Name)</td>
<td>None (Reason) non-visible change</td>
</tr>
</tbody>
</table>

**Objections/Comments:**

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

**SUMMARY:** Install a new bound_plio2 and bound_sss_wired which implement the new pl2 stream output which does edit-directed output in the operators, rather than in bound_plio2.

**REASONS:** Aside from fixing bugs 1490 and 1486, this produces a dramatic speed-improvement (3-5 times in normal cases, much better in certain other ones) in the execution of pl2 "put edit" statements.

**DETAILED PROPOSAL:**

1) plio2_put_util in bound_plio2 has been changed to support the error raising mechanism for quick put edits. plio2_open and display_plio_error have been changed for minor improvements and bug fixes.

2) any_to any .alm in bound_sss_wired has been changed to support quick stream i/o and to fix bug 1490.

3) put_field .alm and pl2_operators .mexp in bound_sss_wired have been changed to fix minor bugs in pl2 stream output and to support quick edit output.

4) a new module, put_format .alm, has been added to bound_sss_wired to implement quick edit output.
Changes made:

Installed a new bound_plio2_ in >ldd>exl>o which provides a more complete error diagnostic when display_pllio_error is invoked following an error in opening or closing a stream file. display_pllio_error was changed.

Installed a new bound_plio2_ and oncode_messages_ in >ldd>exl>o which now use the proper referencing_dir search rules to find an i/o module. plio2_open_ and oncode_messages were changed.

Installed a new bound_pll_operators and a new bound_plio2_ which enables stream oriented edit-directed output to be done in an operator within the user's stack frame, resulting in a major speed improvement. plio2_put_util_, put_field_, pll_operators_, and any_to_any_ were changed and a new module, put_format_, was added.

Installed a new bound_pll_operators_ in >ldd>exl>o fixing bug 1466 in which quick put lists in plli o fail if the request requires the outputting of more than 848 chars. put_field_ was changed.

Installed a new bound_pll_operators_ in >ldd>exl>o fixing bug 1490 in which long conversions in any_to_any_ can clobber program storage. any_to_any_ was changed.
**SUMMARY:**

Make minor fixes to the newly installed `new_message_facility`:

1. Write messages interactively on `user_io`, instead of on `user_output` which can be re-attached.

2. Perform an `iwx_$control: "start"` operation so that wakesups are not lost due to the receipt of a message.
**Title:** Fix bug in profile command

**Author:** R. A. Barnes

---

**Summary:**
profile incorrectly counts the number of instructions for a statement preceding a procedure statement or entry statement. The instructions for the latter are erroneously included in the count for the former. Fix the bug.

**Detailed Proposal:**
profile estimates the cost of a statement by getting the difference between the starting location of this statement and that of the next statement. The problem lies in finding the next statement. Presently, profile uses the next profile entry to find the statement_map entry for the next statement. Since no profile entries are generated for procedure statements and entry statements, this can fail to find the next statement. To fix the bug we take advantage of the fact that statement_map entries are generated sequentially, and simply use the next statement_map entry to get the location of the next statement. (Statement_map entries are generated for procedure statements and entry statements.)
### MULTICS CHANGE REQUEST

<table>
<thead>
<tr>
<th>Ver.</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>74122</td>
<td>MULTICS CHANGE REQUEST</td>
</tr>
<tr>
<td>TITLE:</td>
<td>Show sons volume IU in get_quota</td>
</tr>
<tr>
<td>AUTHOR:</td>
<td>VanVleck</td>
</tr>
<tr>
<td>Planned for System:</td>
<td>H.O</td>
</tr>
<tr>
<td>Fixes Bug Number(s):</td>
<td>not applicable</td>
</tr>
<tr>
<td>Documented in MIB:</td>
<td>not applicable</td>
</tr>
<tr>
<td>Incompatible Changes:</td>
<td>no</td>
</tr>
<tr>
<td>User/Operations-visible Interface Changes:</td>
<td>no</td>
</tr>
<tr>
<td>Coded In:</td>
<td>( )PL/I ( )ALM ( )other-see below</td>
</tr>
<tr>
<td>Performance:</td>
<td>( )better ( )same ( )worse</td>
</tr>
<tr>
<td>DOCUMENTATION CHANGES (specify one or more):</td>
<td></td>
</tr>
<tr>
<td>MPM (vol,sect) get_quota MPAM (sect)</td>
<td></td>
</tr>
<tr>
<td>MOSN (sect)</td>
<td></td>
</tr>
<tr>
<td>PLM's (AN#)</td>
<td></td>
</tr>
<tr>
<td>Info Segs</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>OBJECTIONS/COMMENTS:</td>
<td></td>
</tr>
<tr>
<td>Fix documentation in next MPM update to mention sons' volume ID</td>
<td></td>
</tr>
</tbody>
</table>

**SUMMARY:**

Modify the get_quota command to print the sons volume ID if it is printing the quota for multiple directories (as a result of the star convention) and the sons volume ID differs from the parent directory's sons volume ID.

If long mode is requested, do not print the count of inferior quotas (no longer returned by hcs_quota_read) but do print the sons volume ID.

**REASONS:**

The output of get_quota is misleading if no indication of the volume ID is given, since quota on one volume cannot be transmuted into quota on another.
The `get_quota` command returns information about the secondary storage quota and pages used for a specified directory.

Usage

```
get_quota paths -control_arg-
```

where:

1. **paths** are the names of the directories for which quota information is desired. If one of the paths is `-wd` or `-wdir`, the working directory is used. If no paths are given, the working directory is used. The star convention can be used to obtain quota information about several directories.

2. **control_arg** can be either `-long` or `-lg` to specify that the long form of output is to be used; this control_arg may appear anywhere on the command line.

Notes

The short form of output (the default case) prints the number of pages of quota assigned to the directory and the number of pages used by the segments in that directory and any inferior directories that are charging against that quota. The output is prepared in tabular format, with a total, when more than one pathname is specified. When only one pathname is specified, a single line of output is printed.

The long form of output gives the quota and pages-used information provided in the short output. In addition, it prints information about the number of immediately inferior directories with nonzero quotas. It also shows the time-pace product in units of year-days, along with the date that this number was last updated. Thus, a user can see what secondary storage charges his accounts are accumulating. If the user has inferior directories with nonzero quotas, he has to print this product for all these directories in order to obtain the charge.
TITILE: Reduce load caused by IO daemon.

AUTHOR: VanVleck

Planned for System: not applicable
Fixes Bug Number(s): not applicable
Documented in MTS: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)

- MPM (vol,sect)
- MPAM (sect)
- MOSN (sect)
- MSAN (sect)
- PLMs (AN#)
- daemon
- Info Seans
- Other

OBJECTIONS/COMMENTS:

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY: Change IO daemon software to avoid flushing the cache, AST, and paging device, as was done for the dumper.

REASONS: Because the IO daemon makes a single pass through its input segments, the usual Multics optimizations, which assume that the data will be used repeatedly, are not appropriate.

Making the initiated mode be P for segments being printed also protects users from software bugs in the IO daemon which might accidentally write the user's data.

IMPLICATIONS: none
### MULTICS CHANGE REQUEST

**Ver. 3**
**741022**

**TITLE:** Implement -notify option for dprint.

**AUTHOR:** VanVleck

**PLANNED FOR SYSTEMS:**
- **5.0**
- **5.1**
- **5.2**

**FIXED BUG NUMBER(S):** not applicable

**DOCUMENTED IN MIB:** not applicable

**INCOMPATIBLE CHANGE:** yes

**USER/OPERATIONS-VISIBLE INTERFACE CHANGE:** yes

**CODED IN:** PL/I

**DOCUMENTATION CHANGES (specify one or more):**
- MPAM (sect)
- MSAM (sect)
- PLMs
- Info Segs

**OBJECTIONS/COMMENTS:**
This will use new send_mail_ALM feature to be described in future MCR.

---

**SUMMARY:** Implement the -notify option for dprint and dpunch. If this option is specified, the daemon driver will send a confirming message when the output has been performed which looks like this:

```plaintext
printed >uucp>Multics>Smith>foo.list $1.25 queue 1 prtb 700301
```
giving the pathname of the object and the charge.

Also change the daemon drivers to send a message if a request cannot be performed, whether or not -notify was specified. This message would look like

```
Unable to print PATH. REASON.
```

**REASONS:** The confirmation of the completion of daemon processing has been desired for a long time. Now that we have the secure send_message facility, it is safe for the daemons to use it.

**IMPLICATIONS:** The incompatible change comes from the error messages being sent to the user. Users who do not wish to get these error messages may deny all access or wakeup access on their mailbox to the daemon driver.
(changes to dprint writeup; analogous changes for dpunch)

-send, -nt

Send a confirming message when the requested output has been done of the form

```
printed PATH QUST queue N DEVICE REQUEST NO
```

showing the pathname and the charge for output. See the writeup of the accept_messages command for more information on interuser messages.

(add to Notes)

If a requested output operation cannot be done the daemon process will send a message to the user of the form

```
Unable to print PATH. REASON.
```

Giving the pathname of the object and the reason it could not be output. For example, REASON might be "Entry not found."
Title: Make dprint check logical volume.

Author: VanVleck

Planned for System: 4.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded In: ( ) PL/I ( ) ALM ( ) other—see below
Performance: ( ) better ( ) same ( ) worse

Documentation Changes (specify one or more):
MPM (vol, sect) dprint  MPAM (sect)
MOSN (sect)  MSAM (sect)
PLMs (AN#)
Info Segs
Other

OBJECTIONS/COMMENTS:

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

Summary: Change dprint and dpunch to check that the segment to be processed resides on a public volume and to ask the user if he still wants the request if it does not.

Reasons: The IO daemon drivers will normally have access only to public volumes. A user may set a segment RW to IO.SysDaemon but if the segment resides on a private volume the daemon process will not attach the logical volume before attempting to print the segment.

Implications: Forestall a common user error.
<table>
<thead>
<tr>
<th>Ver. 3</th>
<th>741022</th>
<th>MULTICS CHANGE REQUEST</th>
<th>MCR_1861</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE:</td>
<td>Make dir_info tools handle new attributes.</td>
<td>STATUS:</td>
<td>DATE:</td>
</tr>
<tr>
<td>AUTHOR:</td>
<td>VanVleck</td>
<td>Written:</td>
<td>05/09/76</td>
</tr>
<tr>
<td>Planned for System:</td>
<td>MR 4.0</td>
<td>Expires:</td>
<td>11/09/76</td>
</tr>
<tr>
<td>Fixes Bug Number(s):</td>
<td>not applicable</td>
<td>CATEGORY:</td>
<td>(check one)</td>
</tr>
<tr>
<td>Documented in MTB:</td>
<td>not applicable</td>
<td>1) Lib. Maint. Tools</td>
<td></td>
</tr>
<tr>
<td>Incompatible Change:</td>
<td>no</td>
<td>2) Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td>Coded in:</td>
<td>PL/I</td>
<td>4) 355</td>
<td></td>
</tr>
<tr>
<td>Performance:</td>
<td>better (B)</td>
<td>5) BOS</td>
<td></td>
</tr>
<tr>
<td>Documentation Changes (specify one or more):</td>
<td>same (S)</td>
<td>6) Salvager</td>
<td></td>
</tr>
<tr>
<td>MPM (vol,sect) swg</td>
<td>MPAM (sect)</td>
<td>7) Ring Zero</td>
<td></td>
</tr>
<tr>
<td>MDPN (sect)</td>
<td>MSAM (sect)</td>
<td>8) Ring One</td>
<td></td>
</tr>
<tr>
<td>PLMS (AN#)</td>
<td>9) SysDaemon/Admin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Info Segs</td>
<td>10) Runtime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>11) User Command/Subr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OBJECTIONS/COMMENTS:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

**SUMMARY:** Modify `save_dir_info`, `comp_dir_info`, `list_dir_info`, and `rebuild_dir` to handle the new attributes `sons_level`, `master_dir`, and `tpd`.

**REASONS:** The `dir_info` tools should report changes in these attributes.

**IMPLICATIONS:** none
Title: Fix bug in dump_segment

Author: S. Herbst

Date: Written 10 May 76

Status: A 05/18/76

Expires: 11/18/76

Category (Check One):
- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools
- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools

Document Specify One or More
- MPM (Vol, Sect.)
- PLMS (AN #)
- MOSSN (Sect.)
- MPAM (Sect.)
- MSAM (Sect.)

REASON:

Currently, it fails obtaining the segment's length, which it can get even without status permission.

SUMMARY:

Fix the dump_segment command to work with no status permission on the directory containing the segment.
**TITLE:** Fix bug in enter_abs_request  
**AUTHOR:** S. Herbst  
**STATUS** Written  
**DATE** 10 May 76

|----------------------|-------------------|------------------|

<table>
<thead>
<tr>
<th>MCR 1863</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page 1 of 1</td>
</tr>
</tbody>
</table>

**SUMMARY:**

Change `enter_abs_request` to test that the absentee user `Person_id.Project_id.m` has write access to the `absout` segment if it already exists, or append access to the directory where the `absout` segment is to be created.

**REASONS:**

Currently, when `Person_id.Project_id.m` has null access on this directory, there is no warning when the request is submitted and no `absout` segment is created.
enter_abs_request

-time dtime,  
-tm dtime

Indicates that the user wishes to delay creation of the absentee process until a specified time. It must be followed by a character string representing this time. The format of the deferred time is any character string acceptable to the convert_date_to_binary subroutine (described in Section II of the MPM Subroutines). If the time string contains blanks, it must be enclosed in quotes.

-brief, -bf

Indicates that the message "j already requested." is to be suppressed.

3. -arguments, -ag

Is an optional control argument that indicates that the absentee control segment requires arguments. If present, it must be followed by at least one argument. All arguments following -ag on the command line are taken as arguments to the absentee control segment. Thus -ag, if present, must be the last control argument to the enter_abs_request command.

4. optional_args

Are arguments to the absentee control segment.

Notes

If the pathname of the output segment is not specified, the output of the absentee process is directed to a segment whose pathname is the same as the absentee control segment, except that it has the suffix absout instead of absin. If the pathname of the output segment is specified, the named segment may or may not already exist and it need not have the suffix absout.

The command checks for the existence of the absentee input segment and rejects a request for an absentee process if it is not present.

The effect of specifying the -time option is as if the enter_abs_request command were issued at the deferred time.

See also the descriptions of the commands list_abs_requests and cancel_abs_request for information on displaying and cancelling outstanding absentee requests.

The named segment may or may not already exist.

If the absout segment exists, the absentee user (Person_id.Project_id.m) must have write access to the segment. If the absout segment does not exist, the absentee user requires append permission to the directory in which it is to be created.
**TITLE:** Fix bug in process_overseer

**AUTHOR:** T. Casey

|----------------------|-------------------|------------------|-----------|----------------------|

- Planned for System MR 4.0
- Fixes Bug Number(s): unreported
- Documented in MTB
- User/Operations-visible Interface change? yes/ no
- Incompatible change? yes/ no
- Performance: Better/ Same
- Replaces MCR

<table>
<thead>
<tr>
<th>Interface change?</th>
<th>Yes/ No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance:</th>
<th>Better/ Same</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better</td>
<td>Same</td>
</tr>
</tbody>
</table>

**Objections/Comments:**

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

**SUMMARY:**

Initialize the string containing the initial command line to ("");

**REASONS:**

There are paths through the program that neither set it to an initial command nor initialize it to (""). In these cases, stack garbage is passed to listen_. Since it is a varying string, and stack garbage consists mostly of zeros this early in a process, users hardly ever have problems because of this. However, there are occasions when listen_ gets non-zero garbage and tries to execute it.
Objections/Comments:
Amend documentation to state that it can compare equal only if bit counts are equal

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

1. Allow the compare command to be called as an active function.

2. Add the short name cmp.

REASON:

if [compare a a.old] -then dl a.old
Name: compare, cmp

This function compares the contents of two segments.

Usage:

[compare a b]

where a and b are pathnames, return "true" if and and b are identical, "false" otherwise.
TITLE: Implement last_message active functions

AUTHOR: S. Herbst

STATUS DATE
Written 10 May 76
Expires 15 Aug 76

Category (Check One)
Lib. Maint. Tools
Sys. Maint. Tools
Sys. Prog. Tools

Document Specify One or More

Interface change? yes [x] no
Incompatible change? yes [x] no
Performance: [x] Better [ ] Same
Worse [ ]
Replaces MCR

Objections/Comments:

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Implement the active functions last_message, last_message_time, and last_message_sender, as documented on the following page.

REASONS:

Access to message information by abbrev's and exec_coms. Allows text of message to be parsed and allows easy way to answer a message just received.
Name: last_message, lm

This active function returns the text of the last message received from the send_message command.

Usage: [last_message]

Name: last_message_time, lmt

This active function returns the time that the last message was received from the send_message command.

Usage: [last_message_time]

Name: last_message_sender, lms

This active function returns the sender of the last message received from the send_message command, consisting of the sender's Person_id and Project_id separated by a blank.

Usage: [last_message_sender]

Example: To answer a message that has just been received, type:

send_message [last_message_sender] I agree.
Summary: The -pvname reloader option works in all cases but one. Unfortunately that one case, the reloading of information from a deleted physical volume, is the major reason for having the -pvname option.

Reason: This sad state of affairs results from backup load trying to status the branch (and unfortunately the vtoce) of the object it is reloading. This results in its being told that the volume on which the object exists is not mounted. Deeming this to be confusing, the reloader gives up on that object.

Proposal: When invoked with the -pvname option, the reloader will call nphcs_saelentry_file on such an object and then proceed to reload it (the nphcs_call is needed since saelentry dislikes orphnaning vtoces). This course of action seems safe for the following reasons: 1.) the volume is not mounted and therefore is probably dead, 2.) since the reloader was invoked with the -pvname option the process is asserting to the system that the volume is dead, and 3.) we have a hot copy of the segment in our hands which is likely to be as good as the unreachable copy.
TITLE: Delete obsolete tape programs

AUTHOR: B. Silver

- Code in □ PL/I □ AIM □ other-
- Planned for System MR 4.0
- Fixes Bug Number(s): □
- Documented in MTB
- User/Operations-visible
- Interface change? □ yes □ no
- Incompatible change? □ yes □ no
- Performance: □ Better □ Same □ Worse
- Replaces MCR

Category (Check One):
- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools

Document Specify One or More
- Salvager
- PLMS (AN #)
- SysDaemon/Admin.
- MPAM (Sect.)
- User Cmd/Subr.
- MPM (Vol, Sect.)
- MOSN (Sect.)
- MSAM (Sect.)

Objections/Comments:

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Delete the obsolete programs: reconfigure_tape and tape_meters.

REASONS:

The program reconfigure_tape is no longer called. The tape_meters command no longer works.

IMPLICATIONS:

Clean up the system.

DETAILED PROPOSAL:

Delete reconfigure_tape from bound_reconfigure_. Delete tape_meters from bound_misc_metering_.

None (Reason)
Summary:

Add two new RCP commands, attach_lv and detach_lv. Add features to the existing RCP commands, assign_resource, list_resources, unassign_resource.

Reason:

New commands needed in order to use private logical volumes. New features added to assign_resource and unassign_resource commands in order to make them more useful in an absentee environment. The list_resources command can now list attached private logical volumes.

Implications:

Increase resource control capability. No existing RCP command features have been changed.

Proposal:

See attached documentation. Command features relevant to this MCR are marked in right hand column.
RCP Commands For 4.0

assign_resource

Name: assign_resource, ar

The assign_resource command calls the Resource Control Package (RCP) to assign a resource to the user's process. Currently, only device resources can be assigned. An assigned device still must be attached by a call to some I/O module. If a device is successfully assigned, the name of the device is printed. (If the user requests a specific device that is successfully assigned, the name of the device is not printed unless the user asks for it. See the -device and -long control argument below.)

Usage

assign_resource resource_type -control_arms-

where:

1. resource_type

   specifies the type of resource to be assigned. Currently, only device types may be specified. The "-device" control argument may be used to name a specific device to assign or optional control arguments may be used to specify characteristics of the device to be assigned. The following device type keywords are supported:
   "tape"
   "disk"
   "console"
   "printer"
   "punch"
   "reader"
   "special"
assign_resource

2. control_args

- device XX, -dv XX
  specifies the name of the device that is to be assigned. Any arguments that specify device characteristics are ignored. If the -long control argument (see below) is used in conjunction with this control argument, a message containing the name of the assigned device is printed on the user's terminal, otherwise, no message is printed.

- model n
  specifies the device model number characteristic. Only a device that has this model number is assigned. Its value must be one that is found in the "model" field of a PRPH configuration card.

- track n, -tk n
  specifies the track characteristic of a tape drive. The value may be either 9 or 7. If this control argument and the -volume control argument are not specified, then a track value of 9 is used by default when assigning a tape type device.

- density n, -den n
  specifies the density capability characteristic of a tape drive. There may be more than one instance of this argument. A tape drive is assigned that is capable of being set to all of the specified densities. This command does not set the density of the tape drive. The acceptable values for this argument are:
  200
  556
  800
  1600
assign_resource

-train n, -tn n specifies the print train characteristic of a printer. Its value must be one that is found in the "print train" field of a printer PRPH configuration card.

-line_length n, -ll n specifies the line length of a printer. Its value must be one that is found in the "line length" field of a printer PRPH configuration card. If this field is not specified on a printer configuration card, then this device characteristic will be ignored for this printer.

-volume XX, -vol XX specifies the name of a volume. If possible, the device assigned is one on which this volume has already been placed.

-number n, -nb n specifies the number of resources to assign. All of the resources assigned will have the device characteristics specified by any other arguments passed to this command. If this argument is not specified then one resource will be assigned.

-comment XX, -com XX is a comment string that is displayed to the operator when the resource is assigned. If more than one string is required, the entire string must be in quotes. Only printable ASCII characters are allowed. Any unprintable characters (also tabs or new lines) found in this string will be converted to blanks.

-long, -lg specifies that all of the device characteristics of the assigned device should be printed. If this argument is not supplied, only the name of the assigned device is printed.
RCP Commands For 4.0

assign_resource

-system, -sys
specifies that the user wants to be treated as a system process during this assignment. If this argument is not specified or if the user does not have the appropriate access, then RCP assumes that this assignment is for a nonsystem process.

-wait [n], -wt [n]
specifies that the user wants to wait if the assignment cannot be made at this time because the resource(s) are assigned to some other process. The optional value specifies the maximum number of minutes to wait. If no value is specified then it is assumed that the user wants to wait indefinitely.
RCP Commands For 4.0

attach_lv

Name: attach_lv, alv

The attach_lv command calls the Resource Control Package (RCP) to attach a logical volume. The attaching of a logical volume involves telling the storage system that this volume is attached for this process. A logical volume (unless it is a public logical volume) must be attached for each process that wishes to use it. In order for a logical volume to be attached, it must first be physically mounted. The physical mounting of a logical volume involves mounting all of the physical volumes that comprise the logical volume. This must be done by operations before the logical volume may be attached by any process.

In order to attach a logical volume, a user must have "R/W" access to the logical volume. This access is defined by the Access Control Segment (ACS) that is associated with this logical volume. If there is no ACS for this logical volume, then its access is defined by the default access that was specified when the logical volume was registered.

Usage

attach_lv volume_name

where:

1. volume_name specifies the name of the volume to be attached.
The `detach_lv` command detaches one or more logical volumes that have been attached for the user's process by the Resource Control Package (RCP). The detaching of a logical volume involves telling the storage system that this logical volume is no longer attached for this process. The detaching of a logical volume does not affect the attached/detached state of the logical volume for any other process.

**Usage**

```
detach_lv volume
```

where:

1. `volume` specifies the logical volume or volumes to be detached. A user may detach all logical volumes attached for her/his process by specifying the following keyword:

   "all"

   A user may detach one logical volume by specifying its name.
**RCP Commands For 4.0**

**list_resources**

**Name:** list_resources, lr

The list_resources command lists some or all of the resources that are currently assigned or attached to the calling process by the Resource Control Package (RCP).

**Usage**

```
list_resources -control_args-
```

where:

1. **control_args**
   - may be chosen from the following:
     1. **-long, -lq**
        - specifies that all of the information known about each resource is to be printed. If this argument is not supplied, then only the name is printed.
     2. **-type xx, -tp xx**
        - specifies that only resources of the type XX are to be listed. Currently, this argument must specify a device type.
     3. **-device XX, -dv XX**
        - specifies the name of a device resource to be listed. No other resources are listed.
     4. **-logical_volumes, -lv**
        - specifies that only logical volumes that are currently attached are to be listed.
     5. **-assignments, -asm**
        - specifies that only assigned resources are to be listed.
     6. **-attachments, -atm**
        - specifies that only attached devices are to be listed.
unassign_resource

Name: unassign_resource, ur

The unassign_resource command unassigns one or more resources that have been assigned to the user's process by the Resource Control Package (RCP).

Usage

unassign_resource resource -control_args-

where:

1. resource

   specifies the resource or resources to be unassigned from the user's process. Currently, this means only devices. If a device is attached, it is automatically detached. A user may unassign all devices assigned to his/her process by specifying the following keyword:

   "all"

   A user may unassign one device by specifying its name.

2. control_args

   -comment XX, -com XX

   is a comment string that is displayed to the operator when the resource is unassigned. This comment will be displayed only once, even if several resources are being unassigned. (See the assign_resource command for details about comment strings.)

   -admin, -am

   specifies that a force unassignment is to be done. This argument should be specified by highly privileged users who want to unassign a resource that is assigned to some other process.
Fix linkage_error messages

Fix appropriate module(s) in bound_error_handlers_to again print in linkage_error messages the name being referenced.

Messages now print location of the link itself, which is not helpful. This appears to be a fairly recent bug.
MULTICS CHANGE REQUEST

TITLE: Await VTOCE I/O for truncations

AUTHOR: Bernard Greenberg

Planned for System: MR 4.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: (B)PL/I ( )ALM ( )other—see below
Performance: ( )better ( )same (x)worse

DOCUMENTATION CHANGES (specify one or more)
MPM (vol, sect)  MPAM (sect)
MOSN (sect) MSAM (sect)
PLhs (AN#) 61
Info Segs
Other

CATEGORY (check one)
( )Lib. Maint. Tools
( )Sys. Anal. Tools
( )Sys. Prog. Tools
( )355
( )BOS
( )Salvager
( )Ring Zero
( )Ring One
( )SysDaemon/Admin
( )Runtime
( )User Command/Subr

STATUS | DATE
Written | 05/06/76
Expires | 11/06/76

OACTIONS/COMMENTS:

SUMMARY: VTOCEs are written out today with the assumption that the write will either complete or a salvage will occur. Therefore, addresses are deposited during a truncate operation after the write has started. It is therefore possible for an address to appear on disk in a new VTOCE before it has been removed from the disk copy of the old. Since the VTOC looks completely consistent at this stage, should an unrecoverable crash occur, data is misrouted to the wrong segment.

REASONS: The above policy was designed so as to not have to await VTOCE I/O completion on every truncation. Although the possibility of this misrouting is small, it is a possible path for incorrect data.

Therefore, we propose awaiting the completion of every VTOCE write which involves truncation before depositing.

IMPLICATIONS: Decreased performance of the truncate primitive (when dealing with inactive segments) in real time, although not in virtual time.

DETAILED PROPOSAL: The feature is capable of being inhibited by a highly privileged data base patch.
Multics Change Request

TITLE: Install Phase I of BASIC Consolidation

AUTHOR: M. Weaver

- Coded in: [X]PL/I [X]ALM [ ]other
- Planned for System MR 4.0
- Fixes Bug Number(s) see below
- Documented in MTB
- User/Operations-visible
- Interface change? [ ] yes [X]no
- Incompatible change? [ ]yes [X]no
- Performance: [X]Worse [ ]Better [ ]Same
- Replaces MCR

CATEGORY (CHECK ONE)

Lib. Maint. Tools
Sys. Anal. Tools
Sys. Prog. Tools
Basic Changes

Document
Specify One or More

Status
Written 11 May 76
Expires 25/25/76

Objections/Comments:
Remove MCR 156 from list of installed unapproved software.

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Make the standard basic compiler and runtime have both single and double precision capabilities (although a single program can only have one precision).

Fix basic bugs 060, 061, 062.

REASONS:

Only one basic to maintain. Have figured out a way to make the current ep_basic_operators more efficient, so single precision performance will be only negligibly degraded.

SUMMARY:

Rewrite basic_operators in mexp, with 2 transfer vectors; during execution, operator table pointer will point to correct transfer vector. The single and double precision entry operators are the same (for now) so already compiled extended precision programs will continue to work. The rest of the compiler and runtime will consist of what is now bound_ep_basic and bound_ep_basic_runtime.

The default mode is single precision. The command use_ep_basic is used to change the compiler's mode to double precision. The new command use_ep_basic can be used to change back to single precision mode. use_ep_basic will no longer have to terminate or initiate. The runtime can always handle either precision, except that a program may not call a program of a different precision with numeric or numeric file arguments.

Any performance degradation should be more than made up in the future by optimization involving moving some string handling from pl1 to alm runtime. The extended precision features will all be described in an appendix to the BASIC manual (as well as in ep_basic.info)
This file contains a list of all bugs in the Multics BASIC compiler. Bugs marked with ! have been fixed in the installed basic. Bugs marked with * have been fixed in the basic located in >ldd>exl>.

064 next free number
063 76.05.14 functions which are arguments sometimes blow up, including string functions and functions that reference strings or do I/O
031 74.07.09 basic programs cannot read/write tapes as described in basic manual. basic programs will have to specify the use of the stream_to_record IO module when it becomes available. Tapes are unusable with basic programs until then.
019 74.03.23 rseq option of basic system does not recognize uppercase keywords, keywords with embedded spaces, and requires that all line numbers used in a single program be unique (basic itself requires that line numbers be unique only in single subprogram).

Fixed Bugs
062!76.04.02 after a non-basic program is called with no arguments, the return to the basic program sometimes blows up.
061!75.12.08 qosub returns after on x qosub... cause error
060!75.11.21 sometimes use of complicated expressions in output statements causes succeeding references or functions in the statement to be incorrectly compiled.

(9)
Reverse chronological list of changes to Multics BASIC.

00 May 1976
All changes listed below have been installed in the >unbundled version of basic.

23 April 1976
New compiler and runtime were installed in >ldd>exl>o that fix basic bugs 060, 061, and 062.

26 February 1976
All changes listed below have been installed in the >unbundled version of basic. These include an incompatible file format change. The convert_new_basic_file command is no longer available.

26 February 1976 11400
Fixed bug in >ldd>exl>o basic that prevented -compile option from producing usable object segments.

17 February 1976 13430
Incompatible random file format now used by the runtime in >ldd>exl>o. For more information type help basic_file_changes. If there is a problem and you want to use the installed basic, you can convert your files back to the old format by typing "convert_new_basic_file
filename".
Bug 059 fixed in >ldd>exl>o.

09 September 1975 1445
Bug 058 has been fixed in >ldd>exl>o.

13 August 1975
All runtime changes mentioned below have been installed in the >unbundled version of basic.

24 July 1975 1345
Parts of the runtime recompiled; no functional changes.

15 July 1975 1445
Bug 057 has been fixed in >ldd>exl>o and the runtime physically but not logically modified to aid debugging programs.

30 May 1975 1200
All changes listed below have been installed in the >unbundled version of basic.

07 April 1975 1350
Installed basic runtime in >ldd>exl>o that fixes bug 055.
Installed basic compiler in >ldd>exl>o that doesn't call establish_cleanup_proc.
BASIC is available in extended as well as single precision, which is the default. To get extended precision, type the command "use_ep_basic" with no arguments. Then use basic or basic_system in the usual way; all programs compiled after that will use extended precision arithmetic. To get programs compiled in single precision again, type "use_sp_basic".

The following section discusses some compatibility issues. Single and extended precision programs should not be run at the same time because the numeric arguments and numeric files are not compatible. The command convert_numeric_file is available to convert numeric files from single to extended precision and vice versa.

A new statement type has been added to provide a means of controlling dynamically the number of digits in a numeric value that may be printed as output.

Usage

```plaintext
setdigits formula
```

The value expressed by the formula in the statement is truncated to its integer value. That integer value represents the number of digits to be printed per number by all future PRINT statements until another SETDIGITS statement is executed, or until program execution terminates. One to 19 digits may be specified. This statement applies only to nonintegers; integers of 9 digits or less are printed as such. The tab spacing is adjusted to accommodate the current number length; however, the spacing will never be less than the default.

(END)
APPENDIX D
EXTENDED PRECISION

BASIC is available in extended as well as single precision, which is the default. Programs compiled in extended precision mode do all numeric processing in double precision. These programs should not call or be called by single precision programs because numeric arguments including numeric files are not compatible.

To compile in extended precision mode, type the Multics command "use_ep_basic" with no arguments. All BASIC programs compiled after that will use double precision arithmetic. This effect lasts only for the life of the process or until "use_sp_basic" is typed, which returns the compiler to single precision mode. Note that these commands affect only the compiler; programs of either precision can be run at any time.

To convert numeric files from single to double precision or vice versa, the following Multics command is available.
The `convert_numeric_file` command converts random numeric files used by BASIC programs from single to double precision and vice versa using PL/I conversion rules.

**Usage**

```
convert_numeric_file old_path new_path -control_arg-
```

where

1) `old_path` is the pathname of the file to be converted.
2) `new_path` is the pathname of the converted file.
3) `control_arg` may be one of the following:

   - `double_precision`, `-dp` convert from single to double precision; this is the default.
   - `single_precision`, `-sp` convert from double to single precision.
SUMMARY: Install several changes to the new message facility:

1. Fix bug in mbx_set_max_length that prevents it from working.

2. Fix bug in send_mail_ causing the user's working directory to end up with internal blanks. (that is, the character string pathname of the working directory.)

3. Change send_message in input mode to print a warning when the recipient is deferring messages or is not initialized for accepting messages, and then type "Input:". Lines are then sent a line at a time; no wakeups occur with them. Currently, send_message will not go into input mode in either case.

4. Change the header line printed with a message to include the sender authorization of the message if different from the recipient's authorization. Example:

   "From Herbst.Multics at authorization x:"

5. Change send_message not to attempt to acknowledge a message from a sender with lower authorization, and to print a warning about sending a message to be acknowledged to a user with higher authorization.
TITLE: Make print_aste_ptp more informative

AUTHOR: Bernard Greenberg

Planned for System: 4.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol,sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) 72
Info Segs
Other

OBJECTIONS/COMMENTS:

SUMMARY: The current print_aste_ptp command dumps an Active Segment Table entry in block octal, which is useful, but not as useful as could be to those who have not memorized offsets and lengths of critical fields. Change it to interpret various useful AST fields for such people.

REASONS: Greater utility, and current print_aste_ptp prints a line which always exceeds terminal line length.

IMPLICATIONS: n/a.
pap >udd>m>bsg>s>pap
ASTE for >udd>m>bsg>s>pap at 43060 in sst_seg
044700044560 053724000000 033674063410 034573442672 377004007073 404000000000
000000000000 000000000000 003000003001 043040000002

uid = 034573442672, vtocx 7073 on pvtx 4
max len 255, 3 recs used, 1 in core, cur len 3 (decimal)
Used 05/12/76  1609.5 edt Wed
Modified 05/12/76  0950.4 edt Wed
Par astep = 63410, Son = 0, brother = 53724
Trailer thread = 33674

Flags: usedf aao

<table>
<thead>
<tr>
<th>PAGE</th>
<th>PT</th>
<th>DEVADD</th>
<th>PD COPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>011140401045</td>
<td>34440</td>
<td>2527</td>
</tr>
<tr>
<td>1</td>
<td>000075100041</td>
<td>34442</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>002121100041</td>
<td>34460</td>
<td>2121</td>
</tr>
<tr>
<td>3</td>
<td>377020000001</td>
<td>null</td>
<td></td>
</tr>
</tbody>
</table>
MULTICS CHANGE REQUEST

<table>
<thead>
<tr>
<th>TITLE: Fix 28-5 bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHOR: Bernard Greenberg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Planned for System: MR 4.0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixes Bug Number(s): not applicable</td>
<td></td>
</tr>
<tr>
<td>Documented in MTB: not applicable</td>
<td></td>
</tr>
<tr>
<td>Incompatible Change: no</td>
<td></td>
</tr>
<tr>
<td>User/Operations-visible Interface Change: no</td>
<td></td>
</tr>
<tr>
<td>Coded in: ( )PL/I ( )ALM ( )other-see below</td>
<td></td>
</tr>
<tr>
<td>Performance: ( )better ( )same ( )worse</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOCUMENTATION CHANGES (specify one or more)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM (vol, sect)</td>
<td>MPAM (sect)</td>
</tr>
<tr>
<td>MOSN (sect)</td>
<td>MSAM (sect)</td>
</tr>
<tr>
<td>PLMs (AN#)</td>
<td>61</td>
</tr>
<tr>
<td>Info Segs</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

CATEGORIE (check one)

( )Lib. Maint. Tools
( )Sys. Anal. Tools
( )Sys. Prog. Tools
( )355
( )BOS
( )Salvager
( )Ring Zero
( )Ring One
( )SysDaemon/Admin
( )Runtime
( )User Command/Subr

Written: 05/13/76
Status: A05/R976
Expires: 11/13/76

SUMMARY: Fix a few oversight bugs in MSS 28.5.

REASONS: Reliability and correct operations.

IMPLICATIONS: See reasons.

DETAILED PROPOSAL: Recognize and fix the following:

1. The segment_mover deposits addresses without waiting for notification of VTOCE write complete.
   
   This could cause misrouted data if ESD failed after a crash in this window. Wait for such notification.

2. pc$truncate and pc$cleanup update quota-used cells in segment ASTE's. Stop them from doing so.

3. pc$get_file_map returns a list of addresses to be deposited, culled from semikilled addresses appearing in PTW's. For prewithdrawn segments, including PDS's, this is incorrect. Stop this reportage if entry-hold and dont-null-zero-page bits are on, covering these cases. This nulling caused addresses to be withdrawn against the RPV free store map when such pages are in fact referenced, causing a page creepage.

4. It was mistakenly thought that 28-5 prewithdrawing would obviate the need for the explicit prewithdrawing of the SST name table done by make_segs_paged. In fact, it did not, for the latter, even with fix (3) above, leaves semikilled addresses in the SST Name Table's ASTE, which BOS is incapable of resurrecting. Put back code into...
make_segs_paged to store into the SSTNT, resurrecting the prewithdrawn addresses.

(Installed as 28.5a at MIT on emergency basis)
**MULTICS CHANGE REQUEST**

**TITLE:** Post-Crash Paging Device Address Resurrection.

**AUTHOR:** Bernard Greenberg

**Planned for System:** MR 4.0

**Fixes Bug Number(s):** not applicable

**Documented in MTB:** not applicable

**Incompatible Change:** no

**User/Operations-visible Interface Change:** no

**Coded in:** ( ) PL/I ( ) ALM ( ) other-see below

**Performance:** ( ) better ( ) same ( ) worse

**DOCUMENTATION CHANGES**

<table>
<thead>
<tr>
<th>MPM (vol,sect)</th>
<th>MPAM (sect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSN (sect)</td>
<td>MSAM (sect)</td>
</tr>
<tr>
<td>PLMs (AN#) 61</td>
<td></td>
</tr>
<tr>
<td>Info Segs</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

**OBJECTIONS/COMMENTS:**

**SUMMARY:** After a crash where ESD fails, pages on the paging device which have never been written to disk since assigned to their segment are not reported to the VTOCE of their segment. This phenomenon manifests itself as pages of zeros appearing in segments which have resided on the paging device since truncation/creation. This is particularly bitter in the case of archives and programs which were truncated. Change ALM page control to maintain identifying information (Segment Unique ID and page number) in PD map entries. Change the Physical Volume Salvager to collect pages off of a crashed paging device on a segment by segment basis, resurrecting any such addresses so collected. This will replace the current volume-by-volume PD flush strategy.

**REASONS:** The increased robustness of data kept by the system which will be accrued is clear. Such page losses during ESD-failing crashes have been more than a minor annoyance.

**IMPLICATIONS:** The checksum field of the PD Map entry will have to be removed to make room for the segment UID. This will remove the bulk-store checksumming facility entirely. This facility was introduced in the early days of debugging the Bulk Store controller, and is now felt to be unnecessary in light of the successful debugging of this device.
MULTICS CHANGE REQUEST

TITLE: Delete OPTY config card

AUTHOR: Mike Gray

Planned for System: MR4.0

Fixes Bug Number(s): not applicable

Documented in MTB: not applicable

Incompatible Changes: yes

User/Operations-visible Interface Changes: yes

Code in: ( ) PL/I ( ) ALM ( ) other-see below

Performance: ( ) better ( ) same ( ) worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) ( )
Info Segs ( )
Other ( )

OBJECTIONS/COMMENTS:

SUMMARY:

Remove support for the OPTY config card.

REASONS:

When the DataNets are booted by AS initialization, it will be impossible to provide this function.

IMPLICATIONS:

Few sites (if any) use this feature. If they do, they must convert to use the message coordinator to switch to the operator's terminal.

A survey of Multics sites indicates that GM is the only site currently using the OPTY card.

DETAILED PROPOSAL:

Modify Init_sys_vars to crash the system if it finds an OPTY card, telling the operator to remove it.
### MULTICS CHANGE REQUEST

**TITLE:** Add pointers to special data bases to the SST.

**AUTHOR:** VanVleck

**Planned for System:** MB 4.0

**Fixes Bug Number(s):** not applicable

**Documented in MTE:** not applicable

**Incompatible Change:** no

**User/Operations-visible Interface Change:** no

**Coded in:** ( ) PL/I ( ) ALM ( ) other-see below

**Performance:** ( ) better ( ) same ( ) worse

**DOCUMENTATION CHANGES (specify one or more):**

<table>
<thead>
<tr>
<th>MPM (vol, sect)</th>
<th>MPAM (sect)</th>
<th>MOSN (sect)</th>
<th>MSAM (sect)</th>
<th>PLMS (AN#)</th>
<th>an61</th>
<th>Info Segs</th>
<th>Other</th>
</tr>
</thead>
</table>

**CATEGORY (check one):**


| ( ) Salvager | ( ) Ring Zero | ( ) Ring One | ( ) SysDaemon/Admin | ( ) Runtime |

| ( ) User Command/Subr |

**OBJECTIONS/COMMENTS:**

**SUMMARY:** There are two important storage system tables which reside in internal static rather than in their own segments. These are the AST hash table and the PVIE hold table. Place pointers to these data bases in the SST header.

**REASONS:** These pointers will make manual and automatic debugging much easier.
MULTICS CHANGE REQUEST  

TITLE: Fix bug in status_

AUTHOR: VanVleck

Planned for System: MF 4.0
Fixes Bug Number(s): net applicable
Documented in MTE: net applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: ( ) PL/I ( ) FLE ( ) other-see below
Performance: ( ) better ( ) same ( ) worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect)  MPAM (sect)
MSGM (sect)  MSAM (sect)
PLUs (LAN#)  an61
Info Segs
Other

OBJECTIONS/COMMENTS:

HEADINGS ARE: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY: The status_ primitive does not return the master_dir attribute. Cause it to dc sc.

REASONS: This value is needed by the status command and other system primitives. It is correctly returned by star_.

Page 1 of 1
**MULTICS CHANGE REQUEST**

**TITLE:** Fix bug in star_ and dc_pack.

**AUTHOR:** VanVleck

<table>
<thead>
<tr>
<th>Planned for System:</th>
<th>MR 4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixes Bug number(s):</td>
<td>nct applicable</td>
</tr>
<tr>
<td>Documented in NTE:</td>
<td>nct applicable</td>
</tr>
<tr>
<td>Incompatible Change:</td>
<td>nc</td>
</tr>
<tr>
<td>User/Operations-visible Interface Change:</td>
<td>no</td>
</tr>
<tr>
<td>Coded in:</td>
<td>(B)PL/I ( )AIM ( )other-see below</td>
</tr>
<tr>
<td>Performance:</td>
<td>( )Better ( )same ( )worse</td>
</tr>
</tbody>
</table>

**DOCUMENTATION CHANGES (specify one or more)**

<table>
<thead>
<tr>
<th>APP (vol, sect)</th>
<th>MPAM (sect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSN (sect)</td>
<td>MSAM (sect)</td>
</tr>
<tr>
<td>PLMs (AR#)</td>
<td>an61</td>
</tr>
<tr>
<td>Into Seqs</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
</tbody>
</table>

**OBJECTIONS/COMMENTS:**

**SUMMARY:** Both star_ and dc_pack return garbage for the dates of segments whose VTOCES are unavailable due to I/C error or connection failure.

Return zeroes in this case.

**REASONS:** This garbage can produce anomalous results for directory listing programs and disk_usage_stat.
MULTICS CHANGE REQUEST

<table>
<thead>
<tr>
<th>SUMMARY:</th>
<th>The default for dpunch is -7punch. Change it to be none.</th>
</tr>
</thead>
<tbody>
<tr>
<td>REASONS:</td>
<td>Seven-punch is not used very often; but a very common error is to forget to say -rcc.</td>
</tr>
<tr>
<td>IMPLICATIONS:</td>
<td>Incompatible change. Very few users will be affected, and most of them will be pleased.</td>
</tr>
</tbody>
</table>
Summary

call to ioi_$workspace that reduces the size of the buffer does not always succeed.

Reason

The problem is that ioi_buffer may try to set the max length of the buffer less than the current length.

Detailed Proposal

Truncate the buffer to the new length before setting the max length.
TITLE: Fix bug in ioi usurp_channels

AUTHOR: Noel I. Morris

Reasons:
A bug in ioi usurp_channels causes the system to crash shortly after detaching an I/O disk.

Proposal:
Correct the index supplied to iom_manager$iom_assign when giving the disk channel back to page control.
MULTICS CHANGE REQUEST

TITLE: Fix several auditing inconsistencies

AUTHOR: L. Scheffler

Planned for System: MR4.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)

MPM (vol, sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) AN75
Info Segs
Other

OBJECTIONS/COMMENTS:

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY: Make changes to protection_audit_ and dir_control_error to:

1. ignore write faults for segments with copy switches on in their branches (these are copy-on-writes, not access violations);

2. ignore write faults if the process could have had write permission if it wanted it;

3. consistently audit the proper directory for "attributes" operations in directory control;

4. convert unnecessary syserr calls that print messages on the operator's console to simply log them.

REASONS:

1. & 2. These two cases are not auditable access violations.

3. protection_audit_ currently is inconsistent in choosing which branch to audit for "attributes" operations.

4. The operator can do nothing about the events reported by these messages, and they are not vital to continued system operation.
IMPLICATIONS:

1. & 2. Messages for these irrelevant cases will not clutter the audit log.

3. Audit messages for attributes operations will be correct.

4. The operator will not be needlessly disturbed.

DETAILS:

1. & 2. Add code to protection_audit_$fault (called by fim on illegal procedure and access violation faults) to ignore no-write-permit and not-in-write-bracket faults if the segment's copy switch (in its branch) is on, or if the process could have had write access if it wanted it.

3. Add an argument to protection_audit_access_denied to indicate the type of directory control operation (contents, attributes, or status) being audited. Change the code to audit the parent on attributes or status operations. Change dir_control_error (the only caller of this entry) to pass this information.

4. Change syserr messages for bad calls, and for faults taken in ring 0 with locks locked, to not print on the operator's console.
MULTICS CHANGE REQUEST

TITLE: Change 0355 config card to FNP

AUTHOR: Robert S. Coren

Planned for System: MR 5.0
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: yes
User/Operations-visible Interface Change: yes
Code Int ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)
( ) MPM (vol,sect) ( )MPAM (sect)
( ) MOSN (sect) ( )config ( )MSAM (sect)
( ) PLMs (AN#) ( )Info Segs
( ) Other

OBJECTIONS/COMMENTS:

SUMMARY: Change format of config card describing the front-end processor from

0355 fnp_tag IOM-chan IOM_tag

to

FNP fnp_tag IOM_tag-IOM-chan

REASONS: The name 0355 is obsolete. The placement of the IOM channel number before the IOM tag is non-standard and confusing. Since extensive changes are being made to FNP initialization for MR5.0 (in particular bootloading the FNP from Multics instead of from BOS), this seems like a good time to fix these problems.

IMPLICATIONS: When the hardcore and BOS systems that use the new config card are installed, the config check must be modified, since the 0355 card will no longer work.

DETAILED PROPOSAL: Make the new fnp_init module (which will partially replace tty_init) recognize the new card, and change the BOS subroutine conf355 similarly.
MULTICS CHANGE REQUEST

TITLE: Allow the prelinker to abort operation when any unexpected signal occurs.

AUTHOR: Richard Bratt

Planned for System: not applicable
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Changes: no
User/Operations-visible Interface Changes: no
Coded in (B)PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)
( )MPM (vol,sect)  ( )MPAM (sect)
( )MOSN (sect)  ( )MSAM (sect)
( )PLMs (AN#)  ( )Tools - not yet written.
( )Info Segs
( )Other

OBJECTIONS/COMMENTS:

Summary: Arbitrary unexpected machine faults and hierarchy inconsistencies can cause signals which are not handled by the current prelinker. As a result, answering service startup may fail to complete at sites which support prelinked subsystems. If the prelinker were to detect unexpected signals it could abort its operation, report the error, and return. This would allow the answering service startup to continue. The reason for the failure could then be ascertained and corrected by site personnel while unaffected users worked.

Reason: It is a real drag to have the answering service startup fail.

Proposal: Change the prelinker to establish an "any_other" handler which treats all signals as fatal errors. Provide a new option, "-debug", which causes the prelinker to behave as it does today by establishing a "cleanup" handler in place of the "any_other" handler described above.
TITLE: Fix debug to recognize the brief table

AUTHOR: S. Barr

SUMMARY:

Enough information is contained in the brief table to print source lines, but debug requires a complete symbol table to print source lines.

REASONS:

This is an unnecessary restriction.

DETAILED PROPOSAL:

Remove the check for a complete symbol table in the debug print routine, db_print.
### TITLE
Fix db Parse not to take a fault when blanks are encountered.

### AUTHOR
S. Barr

### SUMMARY
db_parse does a label goto with an illegal value. This is the result of PL1 bug 1497.

(If <expr> then go to <lab>; fails if <expr> contains a function that sets one of its arguments that is also used in evaluating <lab>.)

### PROPOSAL
Change db_parse to avoid this construct.
**Multics Change Request**

**TITLE:** Fix newline bug in abbrev_

**AUTHOR:** Steve Herbst

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>THVV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
</tbody>
</table>

**Status**

- **Written:** 5/18/76
- **Expires:** 11/25/76

**STATUS DATE**

**AUTHOR:** Steve Herbst

- **Coded in:** PL/I
- **Other:** PLI
- **Planned for System:** MR 5.0
- **Fixes Bug Number(s):** 355
- **Documented in MTB:**
- **User/Operations-visible Interface change?** YES
- **Incompatible change?** NO
- **Performance:** Better
- **Replaces MCR**

**DOCUMENTATION CHANGES**

- **Document:** Specify One or More
  - BOS
  - Salvager
  - Ring Zero
  - Ring One
  - SysDaemon/Admin.
  - Runtime
  - User Cmd/Subr.
  - MPAM (Sect.)
  - MSAM (Sect.)

**Objections/Comments:**

- **Use these headings:**
  - Summary of Proposal
  - Reasons for Proposal
  - Implications
  - Detailed Proposal

**SUMMARY:** Fix bug in abbrev$_expanded_line$ causing a trailing newline character in the input line not to appear in the output line.
**TITLE:** Fix the BOS IF Command  
**AUTHOR:** Noel I. Morris

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td>Written</td>
<td>5/17/76</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td>Status</td>
<td>05/25/76</td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td>Expires</td>
<td>11/25/76</td>
</tr>
</tbody>
</table>

**Objections/Comments:**
- MOH
  - Check documentation

**Use these headings:** Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

**Summary:**
The new MOH Section IV (BOS) for release 4.0 describes the IF FLAG mechanism as taking a bit position as an argument. The code was implemented to take a mask as an argument, in a manner similar to the IF SWITCH mechanism.

**Proposal:**
Change the code to match the documentation.
TITLE: Fix Anomaly in BOS WRITE Command

AUTHOR: Noel I. Morris

Reason:
The WRITE command, when invoked with no arguments, does not output a blank line. Instead, it prints "WRITE".

Proposal:
The code in WRITE uses the descriptor for the first argument to the WRITE command to access the rest of the command line. If no arguments are given to the command, the descriptor is invalid. Therefore, it is proposed to change to WRITE command to use the descriptor for the zeroth argument (the command name), advancing it by 6 characters to skip "WRITE".
The format_line active function passes an extra null argument to ioa_. This is not correct.

This extra argument can cause some kinds of control strings to expand incorrectly. For example:

```
{format_line "~/a/ ~"} a b c
```

will generate:

```
/a/ /b/ /c/ //
```

instead of:

```
/a/ /b/ /c/
```

Detailed Proposal

The problem occurs because format_line passes its argument list pointer to ioa_$general_rs, but an active function contains an extra argument for the return string. The program will be changed to construct a new argument list for the ioa_ call.
# Multics Change Request

**TITLE:** Add feature to compare_command_output

**AUTHOR:** S. Herbst

**CATEGORY:** Lib. Maint. Tools

**PLanned for System MR:** 5.0

**Fixes Bug Number(s):**

**Documented in MTB:**

**User/Operations-visible Interface change?** Yes

**Incompatible change?** Yes

**Performance:** Same

**Replaces MCR**

---

**SUMMARY:**

Implement -console_input control argument to the compare_command_output command.

**REASONS:**

Facilitates testing of commands that take input.

Saves having to type console input twice.
Inserts for MPM Command

Control argument:

-console_input, -ci

Direct the same console input to each version of the command in turn. This feature is intended for commands that accept input from the console and saves having to type the same input twice.

Examples:

To test the qedx command with an exec_com segment, use the -console_input control argument:

```
&attach input_line
cmpare_command_output >ssq x my_dirqx -bf -ci
<qedx commands>
q
```
TITLE: Implement send_mail_$access_class

AUTHOR: S. Herbst

SUMMARY:

Implement the new entry point send_mail_$access_class, used to send a message with an access class different from the authorization of the sender.

REASON:

Privileged processes, such as the I/O Daemon, need the ability to send messages to users of lower authorization.

IMPLICATIONS:

New gate entry point mailbox_$wakeup_aim_add_index, also documented.
Name: send_mail_$access_class

This entry point acts like send_mail_ but allows the caller to specify the access class of the message.

Usage: 

dcl send_mail_$access_class
(char(*), char(*), ptr, bit(72) aligned, fixed bin(35));
call send_mail_$access_class
(destination, message, info_ptr, access_class, code);

where:

1. destination same as for send_mail_.
2. message same as for send_mail_.
3. info_ptr same as for send_mail_.
4. access_class is the desired access class of the message.
5. code can be any of the above status codes, or error_table $wakeup_denied if the caller lacks sufficient privileges to send a wakeup to a process of lower authorization.

Notes

Special ring 1 system privileges are required to send a message to a lower or isolated access class.

Special ipc privileges are required to send a wakeup to a lower or isolated access class.
Message Segment PLM AN69

**Name:** mailbox\_wakeup\_aim\_add\_index

This entry point acts like mailbox\_wakeup\_add\_index, sending a message with an optional wakeup, and additionally allows the caller to specify the access class of the message.

**Usage:**
\begin{verbatim}
dcl mailbox\_wakeup\_aim\_add\_index
(fixed bin, ptr, fixed bin, bit(36),
 bit(72) aligned, bit(72)aligned, fixed bin(35));
call mailbox\_wakeup\_aim\_add\_index
(mseg\_index, msg\_ptr, msg\_len, switches, access\_class, id, code);
\end{verbatim}

**Where:**

1. **mseg\_index**
   - is the index of a mailbox. (Input)
2. **msg\_ptr**
   - is a pointer to the message. (Input)
3. **msg\_len**
   - is the length of the message in bits. (Input)
4. **switches are:**
   - **wakeup**
     - ON if a wakeup is to be sent.
   - **mbz**
     - OFF (not used).
   - **always\_add**
     - ON to add the message even if the wakeup could not be sent.
5. **access\_class**
   - is the desired access class of the message (Input)
6. **id**
   - is the identifier of the message. (Output)
7. **code**
   - is any status code returned by
     mailbox\_wakeup\_aim\_add\_index. (output)

**Notes:** Special ring 1 system privileges are required to send a message to a lower or isolated access class.

Special ipc privileges are required to send a wakeup to a lower or isolated access class.
MULTICS CHANGE REQUEST

TITLE: Change $messages_off error message

AUTHOR: S. Herbst

SUMMARY:

Change the error message produced by error_table_$messages_off from:

"User has not initialized for accepting messages"

to:

"User is not accepting messages."

REASON:

Confusion over the meaning of "initialized".

<table>
<thead>
<tr>
<th>Use these headings:</th>
<th>Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objections/Comments:</th>
<th>Info Segs</th>
<th>Other (Name)</th>
<th>AG91</th>
<th>None (Reason)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CODED IN:</td>
<td>FL/I</td>
<td>ALM</td>
<td>other</td>
</tr>
<tr>
<td>EXPLAIN IN DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLANNED FOR SYSTEM MR</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIXES BUG NUMBER(S)</td>
<td>355</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOCUMENTED IN MBT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER/OPERATIONS-VISIBLE</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>INCOMPATIBLE CHANGE?</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>PERFORMANCE:</td>
<td>worse</td>
<td>better</td>
<td>same</td>
</tr>
<tr>
<td>REPLACES MCR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOCUMENTATION CHANGES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPECIFY ONE OR MORE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATUS</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Written</td>
<td>18 May 76</td>
</tr>
<tr>
<td>Status</td>
<td>A 05/25/76</td>
</tr>
<tr>
<td>Expires</td>
<td>11/25/76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CODED IN:</td>
<td>FL/I</td>
<td>ALM</td>
<td>other</td>
</tr>
<tr>
<td>EXPLAIN IN DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLANNED FOR SYSTEM MR</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIXES BUG NUMBER(S)</td>
<td>355</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOCUMENTED IN MBT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER/OPERATIONS-VISIBLE</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>INCOMPATIBLE CHANGE?</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>PERFORMANCE:</td>
<td>worse</td>
<td>better</td>
<td>same</td>
</tr>
<tr>
<td>REPLACES MCR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOCUMENTATION CHANGES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPECIFY ONE OR MORE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STATUS</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Written</td>
<td>18 May 76</td>
</tr>
<tr>
<td>Status</td>
<td>A 05/25/76</td>
</tr>
<tr>
<td>Expires</td>
<td>11/25/76</td>
</tr>
</tbody>
</table>
**TITLE:** Delete tape debug message  

**AUTHOR:** B. Silver

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Status</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td>A 05/25/76</td>
<td>Specifiy One or More</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>355</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salvager</td>
<td></td>
<td>MFM (Vol, Sect.)</td>
</tr>
<tr>
<td>Ring Zero</td>
<td></td>
<td>PLMS (An #)</td>
</tr>
<tr>
<td>Ring One</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SysDaemon/Admin.</td>
<td></td>
<td>M6SN (Sect.)</td>
</tr>
<tr>
<td>Runtime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X User Cmmd/Subr.</td>
<td></td>
<td>MSAM (Sect.)</td>
</tr>
</tbody>
</table>

**SUMMARY:**

A debugging message was put into tape to print IOM status when certain write errors occurred. This message was used to find an MPC firmware error. It is proposed here that this message be deleted.

**REASONS:**

The MPC firmware error has been fixed. This message, however, is still being generated due to some non-fatal write errors. This is confusing to operators.

**IMPLICATIONS:**

Eliminate a confusing and useless message.

**DETAILS:**

Delete the "DEVICE ATTENTION BUG" message from tape_write.pl1.
SUMMARY:

Fix operator console bugs that appear when a system uses an Entry Model Console (EMC). These bugs are:

1. Input line terminated if more than 56 characters are typed.
2. An input sequence "\lower_case_letter" is translated into "upper_case_letter". This is necessary on IBM type consoles, but is unnecessary on an EMC.

REASONS:

It is annoying for operators to keep their input lines to 56 or less characters. In QEDX, the input sequence "\f" is translated to "F". Operators have to type "\f".

IMPLICATIONS:

Have a better operator interface on an EMC.

DETAILS:

Increase size of console input buffer from 14 words to 21 words. Change oc_trans_input_ to ignore escapes before lower case letters. An input sequence of "\f" will remain "\f".
**TITLE:** Install vfile_, with 4.0 changes  
**AUTHOR:** M. Asherman

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>MDM</th>
<th>Status</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
<td>Written</td>
<td>18 May 76</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td>Status</td>
<td>4/05/76</td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td>Expires</td>
<td>11/25/76</td>
</tr>
</tbody>
</table>

- **Document**: Specify One or More  
  - MPM (Vol. Sect.) vfile_, iox_  
  - vfile_status, vfile_stat

**Objectives/Comments:**  
1. Add examples to appropriate documentation
2. Add more overview info. in Ref. Guide

**Use these headings:**  
- Summary of Proposal  
- Reasons for Proposal  
- Implications  
- Detailed Proposal

**SUMMARY:**  
Install new version of vfile_, essentially as described in MTB 269. This implements a number of extensions to indexed files in the areas of synchronization and separate manipulation of keys and records.

**REASONS:**  
The current interface imposes a number of logically unnecessary constraints on users of indexed files. The proposed extensions are easy to implement efficiently.

**IMPLICATIONS:**  
Greater flexibility in structure and access of indexed files, potentially leading to more efficient solutions in many applications. Also more user convenience.

**DETAILED PROPOSAL:**  
See MTB 269.
This I/O module supports I/O from/to files in the storage system. All logical file types are supported.

Entry points in this module are not called directly by users; rather, the module is accessed through the I/O system. See "Multics Input/Output System" and "File Input/Output" in Section V of the MPM Reference Guide for a general description of the I/O system and a discussion of files, respectively.

Attach_Description

The attach description has the following form:

vfile_ path -control_args-

where:
1. path is the absolute or relative pathname of the file.
2. control_args may be chosen from the following:
   -extend specifies extension of the file if it already exists. This control argument is only meaningful with openings for output or input_output; otherwise, it is ignored.
   -share -wtime- allows an indexed file to be open in more than one process at the same time, even though not all openings are for input. (See "Multiple Openings" below.) The wtime, if specified, is the maximum time in seconds that this process will wait to perform an operation on the file. A value of -1 means the process may wait indefinitely. If no wtime is given, a default value of 1 is used.
   -blocked -p- specifies attachment to a blocked file. If a nonempty file exists, p is ignored and may be omitted. Otherwise, p is used to set the maximum record size (bytes).
-no_trunc indicates that a put_chars operation into the middle of an unstructured file (stream_input_output) is permitted, and no truncation is to occur in such cases. Also prevents the truncation of an existing file at open and in stream_input_output openings causes the next byte position to be initially set to beginning of file.

-append in input_output openings, this causes put_chars and write_record operations to add to end of file instead of truncating when the file position is not at end of file. Also the position is initially set to beginning of file, and an existing file is not truncated at open.

-header -h for use with unstructured files, this control argument indicates that a header is expected in an existing file, or is to be created for a new file. If a header is specified, it contains an optional identifying number, which effectively permits user-defined file types. If h is given and the file exists, the file identifier must be equal to h; a new file takes the value of h, if given, as its identifier. The header is maintained and becomes invisible only with the explicit use of this control argument.

-old indicates that a new file is not to be created if an attempt is made to open a nonexisting file for output, input_output, or update.

-ssf restricts the file to a single segment. If specified, an attempt to open a multisegment file or to expand a file beyond a single segment is treated as an error. The file must not be indexed.

-dup_ok indicates that the creation of duplicate keys is to be permitted. The file must be indexed. (See "Duplicate Keys" below.)

The -extend, -append, and -no_trunc control arguments conflict; only one may be specified.
To form the attachment description actually used in the attachment, the pathname is expanded to obtain an absolute pathname.

Opening_and_Access_Requirements

All opening modes are supported. For an existing file, the mode must be compatible with the file type. (See "File Input/Output" in Section V of the W* M Reference Guide.) The mode must be compatible with any control arguments given in the attachment description.

An existing file is not truncated at open if its safety switch is on and its bit count is nonzero.

If the opening is for inout only, only read access is required on the file. In all other cases, rw access is required on the file.

Position_Operation

An additional type of positioning is available with unstructured and blocked files that are open for input, input_output, or update. When the type argument of the iox.$position entry point is 2, this specifies direct positioning to the record or byte whose ordinal position (0, 1, 2, ... ) is given. The zero position is just beyond the file header, if a header is present.

write_Operation

In blocked and sequential files open for update, this operation is supported. Its effect is to append a record to the file or replace the next record, depending on the next record position.

Rewrite_Operation

If the file is a sequential file, the new record must be the same length as the replaced record. If not, the code returned is error_table$_long_record or error_table$_short_record.
In a blocked file, no record may be rewritten with a record whose length exceeds the maximum record length of the file. Attempting to do so causes the code, error_table$long_record, to be returned.

Delete_Operation

If the file is a sequential file, the record is logically deleted, but the space it occupies is not recovered.

Deletions are not supported in blocked files. If the user attempts to delete a record in a blocked file, the code, error_table$no_operation is returned.

Modes_Operation

This operation is not supported.

Control_Operation

The following orders are supported by the vfile_ I/O module.

read_position add_key
seek_head delete_key
set_wait_time get_key
truncate min_block_size
max_rec_len reassign_key
record_status set_file_lock

The five orders in the first column are described below. The remaining orders, documented in the vfile_ I/O module in the M$M Subsystem Writers' Guide, implement various features of indexed files that require somewhat more knowledge of internal file structure than is expected of most users.
The `read_position` order is accepted when the I/O switch is open and attached to a non-indexed file. The operation returns the ordinal position (0, 1, 2, ...) of the next record (byte for unstructured files), and that of the end of file, relative to the file base. The file base is just beyond the header, if a header is present.

For this order, the `info_ptr` argument must point to a structure of the following form:

```plaintext
dcl 1 info based (info_ptr),
  2 next_position fixed(34), /*output*/
  2 last_position fixed(34); /*output*/
```

`seek_head` order

The `seek_head` order is accepted when the I/O switch is open for `keyed_sequential_input` or `keyed_sequential_update`. For this order the `info_ptr` argument must point to a structure of the following form:

```plaintext
dcl 1 info based (info_ptr),
  2 relation_type fixed,
  2 n fixed,
  2 search_key char (0 refer (n));
```

The order locates the first record with a key whose head has the specified relation with the given search_key. The next record position and (for `keyed_sequential_update`) the current record position are set to the record. If no such record exists, the code `error_table$no_record` is returned.
The head of a record's key is the first n characters of the key, the key being extended by blanks if it has fewer than n characters. The allowed values for info.relation_type are:

0  head = search_key
1  head >= search_key
2  head > search_key

set_wait_time

The set_wait_time order is accepted when the I/O switch is open and attached to an indexed file with the -share control argument. For this order the info_ptr argument must point to a structure of the following form:

dcl new_wait_time float based(info_ptr);

This order specifies a limit on the time that the user's process will wait to perform an order when the file is locked by another process. The interpretation of new_wait_time is the same as that described earlier for the wtime limit used with the -share control argument.

truncate

The truncate order is accepted when the I/O switch is attached to a nonindexed file open for input_output or update. The operation truncates the file at the next record (byte for unstructured files). If the next position is undefined, the code error_table$no_record is returned.

No info structure is required for this order.

max_rec_len

The max_rec_len order is accepted when the I/O switch is open and attached to a blocked file. The operation returns the
maximum record length (bytes) of the file. A new maximum length can be set by specifying a nonzero value for the second argument. In this case the file must empty and open for modification, or the code error_table_error no_operation is returned.

For this order the info_ptr argument must point to a structure of the following form:

```plaintext
dcl 1 info based (info_ptr),
  2 old_max_recl fixed(21); /*output*/
  2 new_max_recl fixed(21); /*input*/
```

**Duplicate Keys**

By default, the vfile_ I/O module prevents the user from associating a single key with more than one record in the same indexed file. This restriction is removed when the -dup_ok control argument is used or if the file's statistics indicate that duplicate keys are already present.

Duplicate keys can be created via either the write_record operation or the add_key or record_status control orders. When duplications are permitted, the key for insertion is defined as the key of the current record, if it exists.

With this extension, the notion of an "index entry" becomes more basic than that of a single key in the index. An index entry is an association between a string of characters (key) and a number (record descriptor).

Index entries are ordered by key. Within multiple occurrences of the same key, the order is identical to the order in which the entries were created. A seek_key or seek_head operation locates the first instance of a set of duplicate keys. A write_record operation advances the file position beyond the last instance of the key for insertion, if the key already exists in the index.

The next record position is best thought of as corresponding to the next index entry. Operations that can advance the next record and position (i.e., read_record; rewrite_record; and position, with a type argument of 0) permit one to locate intermediate instances of duplicate keys.
Multiple Openings

It is possible to have or attempt to have multiple openings of the same file, that is, to have two or more open I/O switches attached to the same file. These switches might be in the same process or in different processes. With respect to the effects of multiple openings, the various opening modes can be divided into four classes (explained below). Multiple openings in which the opening modes are in more than one class are invalid as are multiple openings within certain classes. The vfile I/O module prevents some cases of multiple opening. In these cases, error_table$fi!e_busy is returned by the open operation. In cases where an invalid multiple opening does occur, I/O operations will cause unpredictable errors in the processes involved, and the contents of the files may be damaged.

The classes of multiple openings are:

1. Openings for input without the -share control argument. Any number of openings in this class are allowed. The existence of an opening in this class never causes damage to the file. When this class of opening is attempted, the existence of all class 2 and 3 openings and some class 4 openings will be detected for structured files.

2. Openings for output or input_output without the -extend control argument. Only one opening is allowed. The existence of another opening is never detected when this class of opening is attempted. The file is simply replaced by an empty file of the appropriate type. If the file was already open with an opening of any class except class 1, the contents of the new file will probably be damaged.

3. Openings for update without the -share control argument and for output or input_output without the -share control argument and with the -extend control argument. Only one opening of this class is allowed. For structured files, multiple openings within the class are detected. An invalid multiple opening involving an opening of this class and other openings of class 4 may be detected. If not, the only effect is that the class 3 opening locks the file for the entire opening.
Openings with the -share control argument.
Any number of openings of this type are allowed. When a process performs an update on the file, the file is locked. Other processes attempting an operation while the file is locked will wait up to the limit specified by wtime in the -share control argument or from the last set_wait_time order. If the operation is not carried out because of the wtime limit, the code error_table_$file_busy is returned.

There are two codes that pertain only to class 4 openings: error_table_$asynch_deletion and error_table_$asynch_insertion. The first is returned when there is an attempt to reference a record located by the previous operation, but the record has been deleted in some other opening. The second is returned by write_record when a record with the key for insertion (defined by a seek_key operation) has already been inserted (by some other opening).

**Interrupted Openings**

If a process opens a file and terminates without closing the file, the file may be left in an intermediate state that prohibits normal I/O operations on the file. The exception is openings for input only. The details depend on the particular type of file as follows:

1. **Unstructured file.**
   In general, the bit count of the file's last segment will not be properly set. This condition is not detected at subsequent openings, and part of the file's contents may be overwritten or ignored.

2. **Sequential file.**
   In general, certain descriptors in the file and the bit count of the file's last segment will not be properly set. This condition is detected at a subsequent open, and either the file is automatically adjusted or (if the opening is input only) the code error_table_$file_busy is returned.
In general, the file's bit count and record count will not be correct. This condition is detected at a subsequent open, and either the file is automatically adjusted or (if the opening is input only) the code error_table_$file_busy is returned.

4. Indexed file.
In general, the bit counts of the file's segments will not be properly set, and the file contents will be in a complex intermediate state (e.g., a record, but not its key in the index, will be deleted). This situation is detected at a subsequent open or at the beginning of the next operation, if the file is already open with the -share control argument. Unless the opening is for input only, the file is automatically adjusted; otherwise, the code error_table_$file_busy is returned.

When an indexed file is adjusted, the interrupted operation (write_record, rewrite_record, delete_record, etc.), if any, is completed. For rewrite_record, however, the bytes of the record may be incorrect. (Everything else will be correct.) In this case, an error message is printed on the terminal. The user can rewrite or delete the record as required. The completion of an interrupted write operation may also produce an incorrect record, in which case the defective record and its key are automatically deleted from the file.

Any type of file may be properly adjusted with the vfile_adjust command (described in the MPM commands), if an interrupted opening has occurred.

Inconsistent_files

The code error_table_$bad_file (terminal message: "File is not a structured file or is inconsistent") may be returned by operations on structured files. It means that an inconsistency has been detected in the file. Possible causes are:

1. The file is not a structured file of the required type;
2. A program accidentally modified some words in the file.
Obtaining File Information

The type and various statistics of any of the four vfile-supported file structures may be obtained with the vfile_status command or vfile_status subroutine (described in the MPM Commands and Subroutines respectively).
The vfile_ I/O module is documented in this manual and in the MPM Subroutines. The following order calls are for users of indexed files. These orders allow a greater degree of control in the areas of synchronization and separate record/index manipulation. For additional information, see the vfile_ I/O module in the MPM Subroutines.

min_block_size

The min_block_size operation determines the minimum size for blocks of record space that are subsequently allocated by write_record or rewrite_record operations (documented in the iox_subroutine). The specification remains in effect for the duration of the current opening or until another call to this order is issued. The I/O switch must be attached to an indexed file open for output or update.

For this order the info_ptr argument must point to a structure of the following form:

```dcl 1 min_blksz_info based(info_ptr),
2 min_residue fixed bin(21),
2 min_capacity fixed bin(21);
```

where:

1. min_residue specifies the minimum unused capacity of a record block (bytes); i.e., the difference between the record's length and the maximum length it can attain without requiring reallocation. (Input)

2. min_capacity specifies the minimum total record capacity (bytes); i.e., the maximum length that the record can attain without requiring reallocation. (Input)

When the I/O switch is initially opened, both these parameters are set to zero.
The current implementation imposes the following constraints on allocated record blocks:

1. The minimum allocation is eight full words, including two header words for the block length and record length. The minimum non-null record capacity is, therefore, 24 bytes.

2. The size of an allocated block is always an integral number of full words, i.e., a multiple of four bytes.

The formula below gives the allocation size, block_words, used for a write_record or rewrite_record operation with a given buffer length, buff_len:

\[
\text{block_words} = \begin{cases} 
0 & \text{(no allocation if and only if} \\
\text{otherwise,} \\
\max(8, \frac{(\max(\text{buff_len} + \text{min_residue}, \text{min_capacity}) + 3)}{4}) & \end{cases}
\]

record_status

The record_status operation returns information about a specified record in an indexed file, and optionally permits the user to manipulate the record's lock and/or to allocate an empty record.

An argument is provided to permit one to entirely avoid using the index in accessing and creating records (see "Note" below).

The I/O switch must be open and attached to an indexed file. The next record position is not altered or used by this operation. The current record position is always set to the record referenced.
The I/O switch must be open for output or undate in order to lock, unlock or create a record.

For this order the info_ptr argument must point to a structure of the following form:

dcl 1 rs_info based(info_ptr) aligned,
  2 version fixed bin,
  2 flags aligned,
      3 lock_sw bit(1) unal,
      3 unlock_sw bit(1) unal,
      3 create_sw bit(1) unal,
      3 locate_sw bit(1) unal,
      3 mbz1 bit(3?) unal,
    2 record_len fixed bin(21),
    2 max_rec_len fixed bin(21),
    2 record_ptr ptr,
    2 descriptor fixed bin(35),
    2 mbz2 fixed bin;

where:

1. version

   is provided for compatibility with possible future versions of this info structure. The user should set this argument to rs_info_version_1. (Input)

2. lock_sw

   if set to "1"b, an attempt is made to lock the specified record within the wait time limit given at attachment or subsequently set via the set_wait_time order (documented in the MPM Subroutines). Possible error codes are those returned by set_lock$lock as well as the code error_table$no_room_for_lock, which is returned if the allocated record block is too small to contain a lock. (See "Records Locks" below). (Inout)

3. unlock_sw

   if set to "1"b, an attempt is made to unlock the record. Possible error codes are those returned by set_lock$sunlock and the code error_table$no_room_for_lock. If both lock_sw and unlock_sw are set to "1"b, the locking takes place first and determines the resultant error code. (This permits one to clear an invalid lock in a single operation.)
4. create_sw

If set to "1"b, allocates a new record using the record_len and max_rec_len arguments as input parameters. The contents of the record are set to zero, and its lock is set in the same operation, if lock_sw equals "1"b. Depending upon the setting of locate_sw, the new record may be entered into the index. If locate_sw equals "0"b the current key for insertion is added to the index as a key for the new record. Otherwise, no index entry is created and the key for insertion becomes undefined. (Input)

5. locate_sw

"0"b if create_sw also equals "0"b, this indicates that the current record position defines the record of interest. Otherwise, the current key for insertion is used. If the relevant position designator is undefined, the code error_table_$no_record or error_table_$no_key is returned, whichever is appropriate.

"1"b if create_sw equals "0"b, this indicates that the descriptor argument is an input parameter defining the location of the record of interest. When such references are permitted in a shared file, users must observe certain protocols to ensure proper synchronization of access at the record level. Record locks are provided for this purpose. If create_sw equals "1"b, this causes the new record to be created without a key.

6. mbz1 and mbz2

must be set to zero by the user. (Input)

7. record_len

gives the record's length in bytes. (Output) If create_sw equals "1"b this argument is input.

8. max_rec_len

if create_sw equals "1"b this argument is input and overrides any minimum block size
specification that may currently be in effect (see min_block_size order above). The returned value gives the maximum length that the record can attain (bytes) without requiring reallocation. When this argument is used as an input parameter, the resultant maximum record length is the smallest number greater than or equal to max_rec_len that corresponds to an implemented (nonzero) block size. (Output)

9. record_ptr points to the first byte of the allocated record, or is set to null if no allocated record exists. (Output)

10. descriptor is a process independent locator for the specified record. This value is used as an input argument when locate_sw equals "1"b and create_sw equals "0"b. (Output) The actual structure of each descriptor is as follows:

\[
\begin{align*}
\text{dcl 1 descrip_struct based (addr(descriptor))) aligned,} \\
2 \text{ comp_num fixed bin(17) unal,} \\
2 \text{ word_offset bit(18) unal;}
\end{align*}
\]

where:

- **comp_num** is the msf component number of the segment containing the record.

- **word_offset** is the word offset of the block of storage containing the allocated record relative to the base of its file component.

A zero descriptor designates an unallocated (zero-length) record.

Descriptors may also be arguments to the add_key, delete_key, reassign_key, and get_key orders. Notice that at any given time within a single file each record is uniquely located by its descriptor, which remains valid only for the life of a single allocation.
Notes

If locate_sw is set to "1", the resultant current record position moves "outside" of the index in the sense that there is no key associated with the current record. This situation may also arise after using the delete_key operation.

When this is the case, a subsequent rewrite_record or delete_record operation behaves differently from the usual case. The difference is that no corresponding index entry is changed or deleted to reflect the change to the record.

Extreme caution must be exercised when using the control operations that take a descriptor as an input argument, especially in a shared environment. The user is responsible for ensuring that previously obtained descriptors and pointers are still valid when they are used. Also, pains must be taken to maintain the index in a consistent state, i.e., each index entry should designate a valid record if a record reference may be attempted.

get_key

The get_key operation returns both the key and the record descriptor for the next record in an indexed file.

The I/O switch must be open for keyed_sequential_input or keyed_sequential_update. If the next record position is at end of file, the code error_table_send_of_info is returned. If the next record position is undefined, the code error_table_sno_record is returned. The next record position is unchanged, and the current record position is set to the next record if the operation is successful; otherwise, the current record position is set to null.

For this order, the info_ptr argument must point to a structure of the following form:

dcl 1 get_key_info based (info_ptr),
  2 mbz fixed bin,
  2 descriptor fixed bin(35),
  2 key_length fixed bin,
2 key_string char(0 refer(get_key_info.key_length));

where:

1. descriptor is the record locator for the next record. This value may be used as an input argument to the control operations add_key, delete_key, reassign_key, and record_status (see "Notes" below). (Output)

2. key_length is the length of the key at the next record position. (Output)

3. key_string is the next record's key. (Output)

4. mbz must be set to zero by the user. (Input)

Notes

The interpretation of the descriptor argument as a record locator is not mandatory, since the add_key and reassign_key operations permit the user to set the descriptor portion of an index entry to an arbitrary 36 bit value.

In such cases the descriptor itself may be thought of as a one-word record that is read by the get_key operation.

add_key

The add_key operation creates a new index entry with a given key and record descriptor.

The I/O switch must be open for direct_output, direct_update, keyed_sequential_output, or keyed_sequential_update. Current and next record positions are unchanged.

Associations may be formed between any number of keys and a single record via this operation. Duplicate keys may be added if the file is attached with the -dup_key control argument, or if the file already contains duplications; otherwise, the code error_table$key_duplication is returned. (See "Duplicate Keys"
This operation, as well as the delete_key, reassign_key, and get_key orders do not reference the length or contents of a record. This permits one to avoid the use of actual records altogether in any given indexed file.

For this order, the info_ptr argument must point to a structure of the following form:

dcl 1 add_key_info based(info_ptr),
  2 flags aligned,
  3 input_key bit(1) unal,
  3 input_descr bit(1) unal,
  3 mbz bit(34) unal, /*must be zero*/
  2 descriptor fixed bin(35),
  2 key_len fixed bin,
  2 key_string char(0 refer(add_key_info.key_len));

where:

1. input_key (Input)
   "0"b indicates that the current key for insertion is the new key. If this value is undefined, the code error_table$no_key is returned.
   "1"b indicates that the key to be added is the key_string contained in this info structure.

2. input_descr (Input)
   "0"b indicates that the current record defines the new descriptor. If the current record is undefined, the code error_table$no_record is returned.
   "1"b indicates that the user supplied descriptor in this info structure is the new descriptor.

3. descriptor is used only if the variable input_descr is set to "1"b. The descriptor is stored into the index together with its associated key. Any 36-bit quantity may be supplied, although in general this number will have been previously obtained via the record_status or
get_key control operations. Descriptors are used by operations that reference the contents of length of a record, in order to obtain the record's address. (Input)

4. key_len

is the length of the key_string. Keys must be between 0 and 256 characters, inclusive. (Input)

5. key_string

is used only if add_key_info.input_key is set to "1"b. It defines the key to be added to the index with the appropriate record descriptor.

delete_key

The delete_key operation deletes a specified index entry.

The I/O switch must be open for direct_update or keyed_sequential_update. The current and next file positions are left unchanged, with the following exception: if the deleted index entry is at the next record position, then the next record position is advanced to the following index entry, or becomes undefined in direct openings.

For this order the info_ptr argument may be null, or may point to a structure of the following form:

dcl 1 delete_key_info like add_key_info based (info_ptr);

where:

1. input_key

"0"b indicates that the key associated with the current file position defines the key of the index entry that is to be deleted. If current position is undefined or outside the index (e.g., after deleting the current key of the current record), the code error_table$no_key is returned.
"1"b indicates that the user_supplied key_string defines the key of the entry to be deleted. If no such key is found, the code error_table$_$no_key is returned.

2. input_descrip (Input) "0"b indicates that the index entry to be deleted is associated with the current record. If the current record is undefined, the code error_table$_$no_record is returned.

"1"b indicates that the entry to be deleted is associated with the user_supplied descriptor. If no such entry exists, the code error_table$_$no_record is returned.

3. descriptor is used only if delete_key_info.input_descrip equals "1"b. The entry which is deleted is the first whose descriptor matches this value, among those entries with the specified key.

4. key_len same as in add_key above. (Input)

5. key_string if delete_key_info.input_key equals "1"b, this argument defines the key for which the index entry with the specified record descriptor is to be deleted. (Input)

If the info_ptr argument is null, the index entry at the current file position is deleted, i.e., the effect is the same as that of setting both arguments, input_key and input_descrip, to "0"b.

reassign_key

The reassign_key operation causes the descriptor portion of a specified index entry to be replaced with a given value.

The I/O switch must be open for direct_update or keyed_sequential_update. The file position designators are not changed.
For this order the info_ptr argument must point to a structure of the following form:

```plaintext
dcl 1 reassign_key_info based(info_ptr),
  2 flags aligned,
  3 input_key bit(1) unal,
  3 input_old_descrip bit(1) unal,
  3 input_new_descrip bit(1) unal,
  3 mbz bit(33) unal,
  2 old_descrip fixed bin(35),
  2 new_descrip fixed bin(35),
  2 key_len fixed bin,
  2 key_string char(0) refer(reassign_key_info.key_len));
```

where:

1. `input_key` (Input) 
   "0"b indicates that the index entry to be reassigned has as its key the current key for insertion. If undefined the code error_table_Sno_key is returned.
   "1"b indicates that the key_string argument defines the key portion of the index entry to be reassigned. If the key_string is not found in the index, the code error_table_Sno_key is returned.

2. `input_old_descrip` (Input) 
   "0"b indicates that the entry to be changed is associated with the current record. If the current record is undefined, the code error_table_Sno_record is returned.
   "1"b indicates that the old_descrip argument defines the descriptor portion of the index entry to be changed.

3. `input_new_descrip` (Input) 
   "0"b indicates that the specified index entry is to be reassigned to the current record. If the
current record is undefined, the code error_table_no_record is returned.

"1"b indicates that the argument new_descrp is to supply the new value for the descriptor portion of the specified index entry.

4. old_descrp is used only if reassign_key_info.input_old_descrp equals "1"b. The entry that is reassigned is the first whose descriptor matches this value, among those index entries with the specified key. (Input)

5. new_descrp is used only if reassign_key_info.input_new_descrp equals "1". This value replaces the old descriptor of the specified index entry. (Input)

6. key_len as in add_key above. (Input)

7. key_string if reassign_key_info.input_key equals "1"b, this argument defines the key for which the index entry with the specified descriptor is to be reassigned. (Input)

set_file_lock

The set_file_lock order is accepted when the I/O switch is open for output or update and attached to an indexed file with the -share control argument. For this order, the info_ptr argument must point to a structure of the following form:

\[
dcl \text{set_lock_flag} \text{ bit(2) aligned based(info_ptr);}
\]

This operation causes the file to be locked (if possible within the wait-time limit) or unlocked, depending on the user's setting the first bit of info_ptr->set_lock_flag to "1"b or "0"b, respectively.
The possible error codes are those returned by set_lock_slock and set_lock sunlock, excepting the code error_table_invalid_lock_reset, which is not treated as an error.

The second bit of set_lock_flag indicates the class of operations that are to be excluded by locking the file. If "0" only operations that alter the file are excluded, passive operations do not detect this state. Otherwise, all index referencing operations are excluded. In any case, the exclusion only applies to operations outside the current opening.

Record locks

This feature pertains only to indexed files. Record locks provide a basis for synchronizing concurrent access at the individual record level. The setting and clearing of record locks is explicitly controlled by the user via the record_status order.

When the capacity of an allocated record block exceeds its contents by at least four bytes, the last word of the block is treated as a record lock. A nonzero lock identifies the process that set it. The user can ensure that record allocations leave room for a lock by using the min_block_size order with a residue specification of at least four bytes.

All operations that reference the length or contents of an existing record (e.g., seek_key, but not seek_head) also check the record's lock (if one exists). If the record is not locked, the operation proceeds normally. Otherwise, the returned error code reflects the state of the lock, indicating that the contents of the record may be in an inconsistent state. In this case, if the operation does not explicitly involve changing the file, it proceeds normally and the returned code is: error_table_record_busy, if the record is locked by another live process; error_table_slock_is_invalid, if the record's lock is set, but not by an existing process; or error_table_slocked_by_this_process, if the record is locked in the caller's process.

Attempting a rewrite_record or delete_record operation on a record locked by another process has no effect other than to return the code error_table_srecord_busy (file is unchanged). If the lock is invalid, these operations return the code...
error_table: invalid_lock: reset and zero the lock. If the lock was set by the caller, the code returned is error_table: locked_by_this_process. In either case the operation is otherwise successful.

When a record that is locked by the user's process is rewritten, its lock remains set, as long as the minimum block size specification currently in effect leaves enough room for a record_lock.