TO: Distribution
FROM: Joan Archer Scott
DATE: 26 September 75
RE: Multics Change Requests

Enclosed are copies of Multics Change Requests which were approved from 16 July through 31 July 75.
SUMMARY:
Change the format of the linkage section header to include more information.

REASONS:
The linker can be made to run simpler and faster if it can remember information which it needs in the linkage section header.

IMPLICATIONS:
This is an incompatible change that users will have to be told of in advance. However, the only user interface change that will be visible is in unsnapping links. The information needed to handle this feature is reformatted.

DETAILED PROPOSAL:
The format of the linkage section header (after being initialized by the linker) is proposed to be as follows:
The starred items differ from the old declaration. The use of this format will make the following assumptions:

1. The capability for threaded linkage blocks is being removed. The only uses of this in the past were with the bind_archive command (long obsolete) and multiple definition sections implemented as multiple linkage sections with defs-in-link for the last \( N-1 \) sections.

2. The compatibility requirement used to determine which version of linkage section header is being used is the zeroness of the "unused" field. Previously, this was the original linkage section pointer and was guaranteed nonzero.

Note that the only reason the nxt_blk_ptr was needed was to add a definition to a segment which had its original defs in the text section. The new list of definitions, allocated in a second linkage block, was found by threading this new linkage block with the original linkage block. No translators seem to need this generality and there seems to be no reason the system should be cluttered with it.

The programs which currently (should) include the linkdcl include file and are affected by this change are:

```
interpret_link_
get_defptr
link_unsnap_
link_snap
datmk_util_
```

Of these, link_unsnap_ and interpret_link_ are in the user ring. They can be changed to work with either format linkage header. When the hardcore changes are made, they can be changed to expect only the new format.

The addition of the symbol_ptr item to the linkage section header is the first step in eliminating fs_search$get_rel_segment from the system. The last step will require changes to system generation and initialization.
Changes are being made to the linker and its environment. These are

1. The capability for threaded linkage section blocks is being removed, and

2. The location of the original linkage section pointer is being moved.

The threaded linkage blocks were used by the bind_archive command which, although obsolete for a long time, will now no longer be supported at all. The other use of threaded linkage blocks was with multiple definition sections associated with a single text section. This was possible when a segment withdefs in text had a new definition added to it. The new definition was added to a new linkage block (defs in link for this block) which was threaded to the original linkage block.

The new format linkage header is described below. This header does
not contain the block thread but does contain a new item, a pointer to the symbol section for the segment (used in handling *symbol links). The original linkage pointer was moved and made packed to conserve space in the linkage header.

```plaintext
ocl 1 header aligned based,
  2 def_ptr ptr,
  2 symbol_ptr ptr unal,
  2 original_linkage_ptr ptr unal,
  2 unused bit (72),
  2 first_link bit (18) unal,
  2 linkage_section_length bit (18) unal,
  2 text_segment_number bit (18) unal,
  2 static_length bit (18) unal;
```
MULTICS CHANGE REQUEST

TITLE: Retriever Changes
AUTHOR: A. Kohziar

Planned for System: not applicable
Fixed Bug Number(s): unreported
Documented In MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: yes
Coded In: (E)PL/I ( )AM ( )other-see below
Performance: ( )better ( )same ( )worse
Replaces Frm: 829

DOCUMENTATION CHANGES (specify one or more)

PM (vol,sect) MPA'X (sect)
MOSH (sect) NSAM (sect)
PLIs (AM#) 61
Info Segs
Other

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

OBJECTIONS/COMMENTS:

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

SUMMARY:

In order to allow the reloader/retriever to work with AI rules, the placement of an artificial quota of 1 when creating upgraded directories was provided because the backup record which creates the directory cannot contain the quota value. (The subsequent list dir record contains the quota, but there is no guarantee that it will be processed.) Change this mechanism to one which creates upgraded directories security out-of-service and no quota rather than award the 1 page.

REASONS:

Retrievals of upgraded directories today must be issued with the -quota option; otherwise only one segment page will be retrieved before a record_quota_overflow condition occurs. Unfortunately, retrievals with the -quota option allow users to cheat the quota mechanism.

DETAILED PROPOSAL:

Change the (privileged) append function to create security out-of-service upgraded directories rather than award artificial quota. Change the reloader/retriever to turn off security out-of-service whenever quota is restored on upgraded directories (by calling system_privilege_$access_class_check).

IMPLICATIONS:

Retrievals will no longer work correctly in all cases without the quote option. Any process with privilege can move quota down to the upgraded directory.
and place it back into service. Retrievals and reloads will have one more call in the handling of upgraded directories, something that should be negligible.
**SUMMARY:** Some security bugs have cropped up in the new restart fault-signaller mechanism, allowing restart of bad machine conditions in certain cases, and overrunning the PDS with SCU data in certain other cases. Fix these problems.

**REASONS:** These are security holes.

**IMPLICATIONS:** Security. The new restart-fault mechanism, when debugged, is much safer than the old. The initial flurry of bugs currently being experienced is expectable, and should not be taken as a certifiability problem of the new mechanism.

One implication of the proposed solution is a limit on the number of saved sets of SCU-SPL data saved at any one time in a process. To insure adequate room in the PDS for ring zero processing, no more than 350 sets of unstarted, unreleased sets of SCU-SPL data will be allowed. The 351st SCU-SPL set will not be saved, and attempt to restart this data will be signalled as illegal_return.

It is felt that it is virtually impossible to accumulate that much unstarted, unreleased SCU-LPL data in a process unless a deliberate attempt is made to circumvent the normal cleanup mechanism.

**DETAILED PROPOSAL:** See above.
### SUMMARY:

Fix call to internal procedure of page_trace to have correct number of arguments.
TITLE: Fix read status order in message coordinator

AUTHOR: Jerold C. Whitmore

SUMMARY: Change mrd util $read_status to return a valid event channel for every call.

Reasons: The event channel is currently set to zero if there is input ready. This is contrary to the definition of the read_status order call.

Implications: The new IO daemon will no longer get a fatal error when an operator types a bad command.
TITLE: Update the broadcaster_dim to v2pll

AUTHOR: Jerold C. Whitmore

Summary: Update broadcaster_pll to compile and function with the installed version of the pll compiler.

Reasons: The current version uses the old style null pointer comparison and thus will get a fault during attachment with the new ios writearound procedure. Some variable precisions must be changed to allow broadcast_pll to compile correctly with the installed compiler.
### TITLE: Fix Core accounting in pc_abs

### AUTHOR: B. Greenberg

#### SUMMARY: Fix pc_abs to decrement sst.nused when deleting main memory.

#### REASONS: Recent version does not do so, constituting a bug. Effect is to cause bad eligibility computation, i.e., overcommitting main memory.

#### IMPLICATIONS: Better performance.

#### DETAILED PROPOSAL: Fix pc_abs to decrement sst.nused.
**TITLE:** Change initialize.peek_limits error reporting  
**AUTHOR:** B. Greenberg. 

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Date: 8/12/75</th>
<th>Status: A 08/20/75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**-Coded in:** PL/I □ ALM □ other-**
**-Planned for System MR asap**
**-Fixes Bug Number(s)**
**-Documented in MTR**
**-User/Operations-visible Interface change? yes [x] no**
**-Incompatible change? yes [x] no**
**-Performance: [ ] Better [x] Same [ ] Worse**
**-Replaces MCR**

**Objections/Comments:**

**SUMMARY:** initialize.peek_limits, the program which interprets the ASCII data base ring_zero_meter_limits.ASCII, and produces a binary version of the same, fails totally if any segment specified in the former does not exist, and prints out the uninformative and useless message "Syntax error in ASCII segment", which is also incorrect.

Change initialize.peek_limits to report explicitly if a specified segment does not exist, and to keep on going in this case.

**REASONS:** When initialize.peek_limits gives up, the entire conversion is aborted, and a previous, and probably incorrect version of the converted table is used. The error message also gives no clue to the nature of the problem.

**IMPLICATIONS:** Running a Multics will be simplified ever so slightly.

**DETAILED PROPOSAL:** See above.
TITL:\: Change method of virtual CPU time measurement
AUTHOR: Steve Webber

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

DOCUMENTATION CHANGES (specify one or more)

MPM (vol,sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) 60
Info Segs
Other

STATUS DATE
Written 08/12/75
Expires 02/12/76

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

OBJECTIONS/COMMENTS:

SUMMARY:
Change the method of calculating virtual CPU time to correct for double metering and to include segment fault processing time in the virtual overhead.

REASONS:

Fixes inconsistencies, anomalies, and inequities in the current scheme.

IMPLICATIONS:

The charges for certain tasks will change. Since segment fault overhead will no longer be charged to the users, user's charges will be slightly less. However, since double metering bugs are being fixed, less time will be included in the virtual overhead thereby making some charges slightly higher. It is unclear how the balance will work out except that the charges will rarely change by more than a percent or two. At any rate, the charges will be more equitable.
DETAILED PROPOSAL:

Keep a variable in the pds which indicates, if nonzero, that the process is in the midst of a task which should be charged to overhead and not the user. Increment this variable upon entering any overhead task, decrement it on completion of an overhead task and charge the time to the overhead pool only in the variable reaches 0, i.e. no recursive tasks are still active.

Any overhead task which does not complete normally (record quota overflow, segment fault error, etc) will not be charged to the overhead pool, but rather to the user.

Upon exit from ring 0, for whatever reason, the pds variable must be made zero.
SUMMARY:

Add or revise metering tools related to priority scheduler, and tools to set and define work classes.

REASONS:

1. provide new information
2. fix bugs
3. remove redundant information

IMPLICATIONS:

Metering scripts may need to use "print_tuning_parameters" rather than relying on tcm to print tuning parameters.

DETAILED PROPOSAL:

See attached pages.
traffic_control_queue:tcq:

- will print new per-workclass queues.
- will print info about blocked processes if given "-all" arg.
- will print work_class number of each process.
- will print per process delta_cpu and delta_page_faults
  (since last use of tcq) rather than total_cpu and delta_cpu.

The first time it is invoked it will print total_cpu and total_pagefaults per process.

traffic_control_meters:tcm:

- will no longer print any tuning parameters as PTP already
  prints them and more.
- will no longer print idle meters as they are printed by
  total_time_meters (ttm).
- will print response time as seen by processes which have
  received interactive wakeups.
- will print interaction meters and ratios. It is assumed
  that any process which has scheduling parameters = zero
  when made eligible, has interacted. Currently this includes
  tape and printer interactions as well as those resulting
  from tty interactions.

work_class_meters:wcm:

This metering tool is new. It prints on a per-workclass basis
the percent cpu allotted, percent cpu received, and the percent
of eligibilities received.

gen_tcd_card:gtc:

- This command must be modified because the size of the
  header of tc_data has changed.
- gtc will also print the number of words of wired storage
  wasted, if any, by the proposed tcd card.
set_work_class:swc:

New command, requiring hphcs_access. This command will set the work class of processes matching a given processid or matching a given Person.Project.tag.

Usage:

```
swc wc_num id
```

if id is not given it will set the work class of the user of the command. else if id is octal it will assume id is a processid. else it will assume id is a Person.Project.tag.

define_work_classes:dwc:

New command, requiring hphcs_access.

This command will define and set the percentages allotted to the various work classes.

Usage:

```
dwc p1 p2 p3 ....pn
```

where the ith argument is the percent to be allotted to the ith work class. It will not undefine any work class defined in the MGT.
MULTICS CHANGE REQUEST

TITLE: change print_sample_refs to use new object_info_ and iox_

AUTHOR: Ross E. Klinger

Planned for System: not applicable
Fixes Bug Number(s): not applicable
Documented in MCI: not applicable
Incompatible Changes: no
User/Operations-visible Interface Changes: no

Coded in: ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

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

None (reason) no user change

OBJECTIONS/COMMENTS:

SUMMARY: change declarations to use the new object_info_format; change from ios_ to iox_.

REASONS: all system procedures are being changed to use the new object_info_format; all system procedures are being changed to use iox_ whenever possible.

IMPLICATIONS: increased consistency.
## MULTICS CHANGE REQUEST

**Title:** Emergency fixes to tape_ansl

**Author:** Ross E. Klinger

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

### DOCUMENTATION CHANGES (specify one or more)

- **MPH** (vol,sect)  
- **MPAM** (sect)  
- **MOSN** (sect)  
- **MSAM** (sect)  
- **PLMs** (AN#) 57  
- **Info Segs**  
- **Other**

### OBJECTIONS/COMMENTS:

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

**SUMMARY:** These changes were installed on an emergency basis in 26.0 --

1) Fix bug causing control segment to be terminated without invalidating pointer to segment.

2) Eliminate dubious series of calls which strained new tdcn emulation ability.

**REASONS:**

1) The SSS installation of the I/O module contained a bug which resulted in segment fault when a particular series of operations caused the control segment to be terminated without nulling a pointer.

2) The installation of tdcn in 25.10 was unable to handle an admittedly bizarre sequence of calls, resulting in a segment fault, as opposed to the error code which the old tdcn returned.

**IMPLICATIONS:**

I/O module works, and in addition, we found an unreported (different) bug in tdcn.
**Title:** Fix unreported bug in tape_ansl

**Author:** Ross E. Klinger

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

**Summary:** A very obscure bug exists in tape_ansl, such that if a file is opened for sequential_output in generate or create mode, closed without performing any data I/O operations, and the header label group's tape mark is written over the end of tape reflective strip, the I/O switch is not closed.

**Reason:** correct performance

**Implications:** none
SUMMARY: Add several new features to the answering service.

1. If no arguments are given to login, enter, etc., prompt for them.

2. Add a new prelogin request "hello" which will cause the greeting message to be repeated.

3. Add a new prelogin request "slave" which will place a terminal in a state such that the new I/O Daemon can claim it for use as a control terminal.

4. Do not abort user output for warning messages, bump messages, or process termination.

REASONS:

1. Currently the system asks for a password and then responds with an error message. This is foolish.

2. 2741 terminals may receive a gibberish greeting.

3. The way the I/O Daemon currently gets this terminal is by noticing that it is not currently logged in. There is a race condition here which could lead to unpleasant behavior.
4. The output aborting was intended to insure that an ARDS terminal with a full screen would not remain hung forever, with no indication to the user that he was about to be logged out. The abort call doesn’t clear this condition; and its use had the unfortunate effect of scrapping often interesting user output. It is never crucial that the warning message be typed immediately; removing the abort call may delay it for a few seconds. The answering service will continue to abort typeahead when a user process terminates, since the new process is probably unprepared for the old process’s input.

IMPLICATIONS:

Answering service warning messages should really be handled slightly differently by the hardcore. They should bypass (without resetting) the end-of-page-wait condition. It is possible that removing the abort calls might cause a problem with lack of hardcore buffer space. Since the answering service uses the privileged “write force” operation, and since warning messages are fairly short, it is hoped that this will never occur. If it does occur, some users will not get the warning message.
TITLE: Fix bug in resetting a break at an EIS instruction in debug
AUTHOR: Susan Barr

SUMMARY:

The code used to restart a break at an eis instruction causes the zero and negative indicators to be set. This causes incorrect results if the eis instruction was setting those indicators (for a compare, for example).

PROPOSAL:

The indicators will be saved and then restored. The code used to restart a break is kept in the break map header which is in each object segment that has a break. The header has a version number and extra space for the addition of the instructions to save and restore the indicators. Debug will be modified to convert version 2 break map headers to version 3. Currently these self relative instructions are encoded as bit strings in db_break_map and are copied into the break map header. These bit string constants will be removed and the instructions will be assembled into the alm data base db_ext_stat so that the code will be less obscure.

IMPLICATIONS:

This change will not be visible to the users.
TITLE: Set version number before calls to object_info_fn debug
AUTHOR: Susan Barr

CODED IN: [X] PL/I [ ] AIM [ ] OTHER

EXPLAIN IN DETAILED PROPOSAL:

PLANNED FOR SYSTEM MR

FIXES BUG NUMBER(S):

DOCUMENTATION CHANGES:

PLANNED FOR SYSTEM MR

DOCUMENTED IN MTB

USER/OPERATIONS-VISIBLE

INTERFACE CHANGE? [ ] YES [X] NO

INCOMPATIBLE CHANGE? [ ] YES [X] NO

PERFORMANCE: [ ] BETTER [X] SAME [ ] WORSE

REPLACES MCR

REQUESTED DATE

WRITTEN 08/18/75

STATUS

REQUESTED DATE

EXPIRES 02/26/76

SUMMARY:

Fix debug to work with new object_info_fn.

REASONS:

A new object_info_fn was installed that returns a different info structure if the version number is 2. Debug does not set the version number and it occasionally has the value of 2 causing debug to use an incorrect pointer to the breakmap. This causes breaks to be reset using incorrect offsets and legal breaks to be missed.

PROPOSAL:

Set the version number to 1 before each call to object info.

The modules to be changed:

db_print.pll

db_break_map.pll

Note: This has already been submitted as an emergency fix.
### TITLE: Modifying the message segment facility to properly handle large messages

**AUTHOR:** Kenneth T. Pogran (MAC/CSR)

---

**SUMMARY:**

The message segment facility is modified to properly handle large messages. This entails:

1. Changing all fields which deal with the total length of a message in bits from fixed bin (18) or bit (18) to fixed bin (24).
2. To implement (1) above, going from Version 3 to Version 4 message segments.
3. Writing an `ms_convert_v3` procedure to accomplish (2).
4. Changing the PLM (AN69) to indicate that the `message_length` argument is fixed bin (24) rather than fixed bin (18).

**REASONS:**

The length of a message in bits is stored internally within the message segment in a packed, bit (18) field. If a message is greater than 7281 words long, its bit length will not fit into an 18-bit field. Thus, a previous change which upped the maximum message size from 2048 words to 255K words did not do the job intended. In fact, it introduced problems, since, if messages longer than 7281 words are placed into a message segment, they cannot be read out, and an attempt to do so will result in the salvaging of the message segment. To properly handle large messages, the bit length of a message should be stored internally as fixed bin (24).

**IMPLICATIONS:**

1. We will go through the hassle of introducing a new "version" of message segments. The actual change in the content of the message segment will be minuscule of course (unlike the change from Version 2 to Version 3), so the installation should go very smoothly.
2. Large messages (mail) will work properly.
3. The automatic conversion procedure mentioned in S3 and I1 above ensure that existing Version 3 message segments will be upgraded to Version 4 as they are used.
**Multics Change Request**

**Title:** New ready message procedure

**Author:** Kenneth T. Pogran (MAC/CSR)

### Category (Check One)
- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools
- 355
- Document
- Specify One or More

### Category (Specify One or More)
- Documentation Changes
- Library Maint.
- System Analysis
- System Programming
- Other

### Objections/Comments:
- None (Reason)
  - No interface change.

---

**SUMMARY:** A new version of `print_ready_message` is installed. The new version performs exactly the same functions as the old version, but replaces the call to `ioa` to print out the ready message with picture conversions and a call to `iox_put_chars`.

**REASONS:** This new version of `print_ready_message` saves two milliseconds per call over the old version. My measurements indicate that this is about 25 per cent of the actual processing time of the old version (determined by subtracting the cost of calling "nothing" from the cost of calling "ready").

**IMPLICATIONS:**

With my current implementation there is one visible change in the ready message, which I consider to be desirable: The pictures I chose to convert the time of day print out all four digits of the time, instead of suppressing leading zeros as the current ready message does. Thus, a ready message printed at three minutes after midnight will appear with a more conventional-looking time of day, thus:

```
0003
```

instead of the current:

```
3
```

This, of course, can be easily changed back to the present, confusing (in my opinion) form if this minor change is deemed undesirable.
TITLE: Impose 7280 word limit on length of messages in message segments

AUTHOR: Kenneth T. Pogran (MAC/CSR)

SUMMARY: Impose limit of 7280 words on maximum size of a message placed in a Ring 1 message segment (either queue or mailbox).

REASONS: With MCR 1217, an old limitation of 2048 words on the maximum size of a message in a message segment was removed. As it turns out, the message segment primitives store the length of a message internally in bit (18) or fixed bin (18) fields. A message longer than $2^{18}$ bits (7281 words) may be added "properly" to a message segment (at least, nothing seems to go wrong), but it cannot be read or deleted, as the message segment facility is tripped up by the folly of its 18-bit arithmetic. A companion MCR proposes that these fixed bin (18) and bit (18) fields be changed to fixed bin (24); meanwhile, as a temporary fix, a practical limit on the size of messages placed in a message segment will be imposed.

IMPLICATIONS: "Overly long" messages, which now cause the message segment facility to trip over its feet and salvage the message segment when it encounters them, will be prohibited until the message segment facility can be modified to handle them properly.

DETAILED PROPOSAL: 
`mseg_data_smax_message_size`, in `bound_mseg_`, will be changed from 261120 to 7280.

NOTE: Submitted at MIT as an emergency fix.
**TITLE:** Fix bug in GCOS Simulator  
**AUTHOR:** T. Casey  

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

<table>
<thead>
<tr>
<th>DOCUMENTATION CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Specify One or More</td>
</tr>
<tr>
<td>MPM (Vol, Sect.)</td>
</tr>
<tr>
<td>PLMS (AN #)</td>
</tr>
<tr>
<td>MOSN (Sect.)</td>
</tr>
<tr>
<td>MPAM (Sect.)</td>
</tr>
<tr>
<td>MSAM (Sect.)</td>
</tr>
</tbody>
</table>

**Objections/Comments:**

**SUMMARY:**

Change the way the GCOS Simulator modifies the machine conditions when returning to the slave program after a MME or other fault, to be compatible with the latest method of verifying changed machine conditions.

**REASONS:**

Current simulator sometimes fails, attempting to restart illegal machine conditions.
TITLE: Fix bug in ed_mgt

AUTHOR: T. Casey

- Coded in X PL/I □ AIM □ other-
- explain in DETAILED PROPOSAL
- Planned for System MR 3.0
- Fixes Bug Number(s) unreported
- Documented in MTB

Category (Check One)

Lib. Maint. Tools
Sys. Anal. Tools
Sys. Prog. Tools

355

Document Specify One or More

BOS
Salvager
Ring Zero
Ring One
SysDaemon/Admin.
Runtime
User Cmmd/Subr.

Interface change? □ yes X no
Incompatible change? □ yes □ no
Better □ Same □ Worse

SUMMARY:

A line of code accidentally left out of ed_mgt causes it to operate continually as if the MGT had just been converted from version 1 to version 2. The missing line will be added.

REASONS:

The bug causes some inconvenience, forcing the user to do some additional typing to accomplish certain edit operations.

IMPLICATIONS:

It will operate as designed and documented.