Enclosed are copies of Multics Change Requests which were approved from 1 April 75 to 15 April 75.
# Multics Change Request

<table>
<thead>
<tr>
<th>TITLE: Fix bugs in 25-3 with respect to variable size lots</th>
<th>STATUS</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHOR: Webber, S.</td>
<td>Written</td>
<td>03/25/75</td>
</tr>
<tr>
<td>Category (Check One)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lib. Maint. Tools</td>
<td>Expires</td>
<td>06/01/75</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOCUMENTATION CHANGES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coded in [X] PL/I [X] AIM [ ] other-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>explain in DETAILED PROPOSAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned for System MR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Fixes Bug Number(s) Unreported</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Documented in MTB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-User/Operations-visible</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Incompatible change? [X] yes [ ] no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Performance: [ ] Better [X] Same</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Replaces MCR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expires 06/01/75</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SUMMARY OF PROPOSAL:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fix bugs in lot_maintainer and fs_search introduced in 25-3 having to do with variable size linkage offset tables.

**REASONS AND JUSTIFICATION:**

The program lot_maintainer was using an obsolete linkage section header.

The program fs_search was not remapping an error code properly.

**NOTE:** Installed in 25-3 on emergency basis.
TITLE: Shorten int entry and int_entry_desc sequences in pll_operators_

AUTHOR: R. A. Barnes

SUMMARY:

Remove one instruction from the int_entry and int_entry_desc sequence in pll_operators_.

REASONS:

Speed up non-quick internal procedure calls.

IMPLICATIONS:

Programs that violate PL/I rules by fashioning an entry variable out of the address of an internal entry or procedure plus a null display pointer will cause a fault in pll_operators_. These programs were always in error.

DETAILED PROPOSAL:

See MTB 176
**SUMMARY:**

Install `initialize_builtin`, a tool which is necessary to create the PL/I compiler.

**REASONS:**

Honeywell has a contract which specifies that the system must be complete, i.e. able to reproduce itself. `initialize_builtin`'s omission from the tools library meant that the PL/I compiler could not be recreated without the experimental library.

**DETAILED PROPOSAL:**

`initialize_builtin` is used to initialize pli_data, a data segment that is a component of the semantic translator used in processing builtins.
**Name:** initialize_builtin

The execution of this program initializes an array of n structures in pll_data, where n is the number of names by which builtin functions are known. Each structure contains information on how the builtin function referenced by that name is to be handled by the compiler.

**Usage:** initialize_builtin

To initialize the builtin_name array in pll_data the following steps are performed:

1) extract the source program pll_data.alm.
2) assemble pll_data.alm in the working directory in which initialize_builtin is to be executed.
3) execute initialize_builtin.
4) replace the newly-created pll_data object segment in its archive.
Fix bugs in tape_. A bug in tape_labeler_ causes the "rewind" order command to function incorrectly. A bug in tape_write_ causes data moves to be done with a CSL instruction rather than the faster MLR instruction.

IMPLICATIONS:

Reading a tape after issuing a "rewind" order command will now be performed correctly. All tape writing via tape_ will be more efficient.

DETAILED PROPOSAL:

Change tape_labeler_ to reset error information whenever it initializes its data bases. Change tape_write_ to use an aligned rather than an unaligned based array for moving data.
TITLE: Check for out_of_service privilege in find

AUTHOR: A. Kobziar

SUMMARY:

If a known directory has been deactivated, find checks whether it is security out_of_service, but does not check if out_service privilege is on. Change find to make this check when deciding to return an error code.
SUMMARY:
Convert nothing.pl to Version 2 PL/I.

REASON:
Version 1 PL/I programs can no longer be traced. "nothing" is a useful sink for calls, and tracing it can catch such calls.

IMPLICATIONS:
nothing will be useful again.

DETAILED PROPOSAL:
No source changes; recompilation only.
TITLE: Install an assign that sets the support bit

AUTHOR: R. A. Barnes

SUMMARY:
assign_ should set the support bit in its stack frame.

REASONS:
assign_ is a support routine called by such programs as pl1, plio2, debug, etc., to do conversions when the characteristics of source and target are not known until runtime. assign_ calls any_to_any_ to actually do the conversion. If a condition is signalled, we would want the default handler not to say that the condition occurred in assign_, but rather in its most recent non-support ancestor.
**SUMMARY:**

Change propagate to call itself recursively only when there is more than one level of synonyms.

**REASONS:**

Currently, a recursive call is made for every IOCB, including one for the IOCB on which propagate was called. The proposed change saves one recursive call in every case and one for each first level synonym.
**SUMMARY:**
Implement the new `iox_$move_attach` as described in MTB-061.

**REASONS:**
There is a need to be able to transfer an attachment from one IOCB to another without calling the dim's detach entry.

**DETAILED PROPOSAL:**
Add the module `move_attach` to the I/O system and add the entry to `iox_`.

**DOCUMENTATION:** See attached page.
**Name:** iox_$move_attach

This entry moves an attachment from one I/O switch, \( s_1 \), to another I/O switch, \( s_2 \). The switch \( s_2 \) must be in the detached state when the entry is called. If it is not, the code error_table_$not_detached is returned and no change is made to \( s_1 \) or \( s_2 \). If \( s_1 \) is in the detached state, the code error_table_$not_attached is returned and no change is made to \( s_1 \) or \( s_2 \).

Moving the attachment moves the attach description and open description of \( s_1 \) to \( s_2 \). All pointer values and entry values are copied from the control block of \( s_1 \) to the control block of \( s_2 \). (These values are listed in the Subsystem Writer's Guide Section, the I/O Control Block). Attach and open data blocks maintained by the I/O Module (if \( s_1 \) is attached) are not affected. Finally, \( s_1 \) is set to the detached state and iox_$propagate is called for both \( s_1 \) and \( s_2 \).

**Usage:**

```plaintext
declare iox_$move_attach entry (ptr,ptr,fixed(35));
call iox_$move_attach (iocb_ptr_1,iocb_ptr_2,code);
```

1. `iocb_ptr_1` points to the control block (Input) for \( s_1 \)
2. `iocb_ptr_2` points to the control block (Input) for \( s_2 \)
3. `code` is a system status code (Output)
TITLE: Install New Graphic System

AUTHOR: Bob May (PHX) for Chris Tavares (FSO)

STATUS: Written 02/12/75
STATUS: Expires 08/12/75

Coded in [X] PL/I [X] AIM [X] other
- Planned for System MR 2.1
- Fixes Bug Number(s) na
- Documented in MTR 137
- User/Operations-visible [X]
- Interface change? [X] yes [ ] no
- Performance: [X] Better [ ] Same [ ] Worse
- Replaces MCR

Objections/Comments: A plan for reviewing and publishing the updated GUS and for auditing code is needed.

Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)

SUMMARY: The new Multics Graphic System replaces an existing, but unsupported, version in >sss. It provides full functionality for a variety of terminals including the ARDG, IMIAC, CalComp and Tektronix devices. An initial release was required for MR 2.0.

DETAILED PROPOSAL: The Release 2.1 (IQ75) is intended to do the following:

- Delete the obsoleted software from >sss. This includes the following:
  a) bound_graphics
  b) bound_ism
  c) gr_print (in bound_debug). Convert the obsoleted call from gr_print to gr_print_in unbundled.

- Clean up the include files and several minor bug fixes.

- Add a new source language compiler, compile_gdt, to the system. This command is used to compile binary tables for driving the various graphic devices. Source for this must be suffixed with "._gdt". compile_gdt translates the source to AIM and then automatically calls the assembler.

IMPLICATIONS: No users currently keep graphic databases in the old format. Those users who have segments of old Multics Graphics Code (graphics "printfiles") must convert these segments to the new format. The command, tmgc, will do this for them. The subroutines, gui and plot, are replaced with incompatible interfaces. Conversion to the new interface is necessary but minor. A survey taken at MIT produced very little objection to the changes.
### SUMMARY:

It is currently impossible to guarantee that a given process will succeed in reading through a list of messages in a given message segment if some other process is able to delete messages from that same message segment. It is proposed that this deficiency be corrected.

### Reasons:

If a message is deleted from one of the IO daemon queues at a critical moment, the IO coordinator can be misled to believe that there are no more requests in that queue to be processed. This situation lasts only temporarily, however. The IO coordinator will begin again reading messages from the head of the queue, thereby possibly repeating a small number of requests. Soon, however, when the changes to the IO daemon currently in progress have been installed, this problem will become more severe. A much larger number of requests may be repeated due to the fact that requests of different access classes may, at times, be placed on a "waiting" list and hence remain in the queues.

The problem described above is really a general problem. It therefore most likely affects the absentee facility as well as any other subsystem which attempts to use message segments as a queueing mechanism.

### Implications:

Use of message segments will be more consistent and deterministic.
Detailed Proposal: The method of "incremental" message reading supported by the message segment facility depends on the existence of the last read message in order to find the next message to be read. The caller of the message segment facility supplies the message ID of the previously read message. If this message has been deleted, an error code is returned which is the same as that returned when the end of the message list is reached. Hence, the caller cannot distinguish between reading the end of the list or having a single message deleted out from under him!

To correct this problem, advantage will be taken of the fact that a message ID is simply a clock time and that the message list is kept in chronological order. If a call is made to read the next message after message X, an attempt will be made to locate message X as usual. If message X does not exist, then the message list will be searched from the beginning until a message with an ID greater than X is found. (A similar approach will be taken for backward incremental reading.)
TITLE: Modifications to net_connect

AUTHOR: Kenneth T. Pogran

SUMMARY: The following changes are made to net_connect:

1. A tracing facility is added to net_connect. Calling the entry point
   net_connect$tracing_on enables tracing in the user's process; calling
   net_connect$tracing_off disables tracing. The default is off.

2. net_connect returns to the user the proper error code
   (error_table$ncp_error) for sockets placed into the "Broken" (13) state
   by the NCP due to an NCP Software Error.

3. Entry points are added to net_connect for the establishment of
   "privileged" connections -- connections from sockets whose 24-bit "User ID"
   (see NUS 6.2.1) is other than that of the calling process. Use of
   these entries is restricted to those users with access to netp.

4. Three new connection modes are added, corresponding to the present "ICP",
   "Connect", and "Listen" modes, for which the record of the connection
   attempt is not automatically deleted by net_connect in the event the
   connection attempt is unsuccessful. Instead, the user must call
   net_connect$conclude_connection (or net_connect$conclude_priv_connection,
   as appropriate) to determine the outcome of the connection attempt.
   Operation of net_connect with the present connection modes is not
   changed.

5. Calls to ipc$decl_ev_wait_chn are replaced with calls to
   ipc$delete_ev_chn where appropriate.

6. The internal structure of net_connect is improved.

REASONS: "The net_connect module provides a convenient means for a user to establish
a connection to a socket or pair of sockets at any Network host."
   -- NUS, Sect. 6.3.2

1. Occasionally the connection process goes astray, with one of the
   sockets ending up in an unanticipated state. This can lead to
   net_connect returning to the user the code error_table$net_icp_bad_state
"Initial connection socket is in an improper state." When this error occurs consistently when a user attempts a particular network connection, it is an indication that something is wrong: for example, the foreign host is not performing the Initial Connection Protocol properly. The tracing feature of net_connect, when enabled, will print a message at each stage of the connection process, informing the user of the actual states of the sockets involved. With this information it is usually possible to determine what has gone wrong in the connection process.

2. net_connect did not previously handle the transition of a socket into the "Broken" state correctly. It returned error_table $net_icp_bad_state, implying that something had gone wrong with the connection attempt, rather than with the socket itself.

3. Upcoming uses of net_connect in Server process applications, such as Jerry Rudisin's experimental RSEXEC Server and, in particular, the redesigned, more modular logger for the Answering Service process, require this capability.

4. Presently, a status code indicating the outcome of the connection attempt is passed as the event message of the IPC wakeup sent by net_connect. Designers of Server applications using net_connect (especially logger, noted in (3) above) would prefer not to rely on the integrity of the event message, instead using the wakeup as an indication that net_connect $conclude_connection should be called to determine the outcome of the connection attempt.

5. When net_connect was originally implemented, there were bugs in both ipc and in the NCP which together resulted in faults in the user's process when net_connect deleted event channels at what should have been perfectly legitimate but, due to the bugs, unfortuitous moments. The interim solution was to never delete event channels, but to turn them into event wait channels which were never "waited" upon. As both bugs have now been fixed, net_connect can be changed to once more do the "right" thing.

6. As more features were introduced, the need for better internal structure in net_connect became apparent.

DETAILED PROPOSAL:

1. After checking the state of all three sockets upon receiving a socket-state-change event call wakeup, net_connect will print a message (if tracing is enabled) of the form:

   net_connect: Trace U=n, S=s, t, u

   where n is the pin number of the ICP socket, and s, t, and u are the states of the ICP pin, read, and write sockets, respectively.

3. The following entry points are added:

   net_connect $initiate_priv_connection
   net_connect $conclude_priv_connection
   net_connect $abort_priv_connection

   Use of these entries is restricted to those users with access to the gate netp, as netp $ncp_priv_activate is used to activate the relevant sockets.
### SUMMARY:

Several new modes and orders have been designed with MCS. These represent changes in the handling of some old modes, removal of others, and the addition of several new modes and orders. Also the removal of the special printer on and printer off characters (\006 and \025) is proposed.

### REASONS:

These new features and capabilities are supported by MCS and should be available to users. Also, since many of these changes have been suggested for a long time, now seems like a good time to implement them.

### IMPLICATIONS:

Certain user and system programs will have to be modified as a result of dropping \006 and \025, however order calls are available now to eliminate the need for these special characters.
DETAILED PROPOSAL:

1) The mode "hndlquit" has been improved. Besides not doing an automatic reset-read, it no longer does an automatic reset-write, giving the user complete control of his terminal when the QUIT button is hit.

2) Remove "writew" mode. This is the write wait mode and was designed for the on-line T&D group so that a T&D program would know how much data had been written to the terminal when the QUIT button was pressed, so that it could retypen any lost output. It does not work now and is not used. The new "hndlquit" mode is better suited to this purpose.

3) Several new orders were invented to enable the answering service to read answer backs and change terminal types appropriately. These new orders are:

   a) wru  initiate the reading of answer backs; may only be issued by the hproc for the line, i.e., the answering service.

   b) store_id  save the answer back id for later use by the user process.

   c) set_type  change the terminal type of the line. This affects only the type of translation and the new-line delay timings used.

4) Drop the special meaning of \006 and \025. These characters are currently used by some programs to turn on and off the printer of a particular terminal, however these are nonstandard definitions of these characters and with the addition of echoplex, these characters are not sufficient to control the action of the keyboard-printer connections at the users terminal. Two orders "printer_on" and "printer_off" have existed for a long time and were intended to replace the use of \006 and \025. The removal of support for these nonstandard characters will force the issue. Note that the ARPA Net software for Multics does not recognize any special significance of these characters and this has not caused any special hardships.

5) Add a new mode "vertsp"; if on it indicates that the terminal is capable of vertical spacing actions, i.e., form feed and vertical tab, and these characters will be sent to the terminal. Otherwise they will be escaped.
This file contains version 2 PL/I BUGS, PLANNED IMPROVEMENTS, PLANNED LANGUAGE FEATURES, UNDIAGNOSED PROBLEMS, and FURTHER INFORMATION.

BUGS marked with * have been fixed in the experimental compiler located in >ldd> with * will in general, exist in both the installed compiler and the experimental compiler.

1337 next free number
1336 75.03.27 eew based_char1 = translate(auto_char1, const1, const2); sometimes gets FATAL ERROR 310. (isn 7038).
1335 75.03.24 rhs abs(unaligned) fails inside an expression because an unaligned temporary is used.
1334 75.03.19 rhs call foo(arith_array), where foo is declared with a char(*), varying input parameter, fails. Target string gets wrong length.
1333 75.03.18 rab char32_varying = fixed_dec31; fails. Whole string temp not copied.
1332 75.03.15 rhs call foo((i)); where i is fixed bin and foo expects (*) fixed gets FATAL ERROR 336 with table option. Compiler also mishand promotion of i to (**) fixed bin.
1331 75.03.14 rhs if one member of a structure is passed as arg, and the structuring is set in an aggregate assignment stmt after that, and the struct auto or static, another member of the structure may get an err WARNING 307 under rare circumstances.
1330 75.03.13 rhs dcl ptr builtin call pointer(...); is not diagnosed (isn 7036).
1329 75.03.10 eew compiler should diagnose attempts to initialize static ptrs to other than null or null() (mpfr 8096).
1328 75.03.06 eew char32_aligned = substr(varying, 4, 1); may fail due to lack of necessary masking.
1327 75.03.05 eew dcl a(2) fixed bin int static init(1,2,3); does not get ERROR (ckD674).
1326 75.03.04 dsl atan(1/0) fails sometimes, returning negative of correct value. (mpfr 7768)
1325 75.02.14 eew complex_flt_bin = real_scaled_fixbin; fails.
1324 75.02.14 rhs in the general case of an error raised in record-directed i/o the error, not transmit, condition is raised.
1323 75.01.10 rhs allocate stmt of struct with refer extents followed by 2 uses of hbound on contained array with refer extents can get ERROR 313.
1322 75.02.03 rab index(substr(s;i),"1"n) where i has precision > 18 may fail.
1321 75.02.03 rhs bit | char | copy | high | low | hbound | lbound | adrel | baseno | baseptr convert | line | offset | nageno | str | rel | stat | valid builtins req scalar arguments but no check is made.
1320 75.01.31 rhs add | subtract | multiply | divide | fixed | binary | float | decimal bit | character | and precision builtins do not check for negative constant lengths or precisions.
1319 75.01.30 eew char32_aligned = char32_aligned; can fail.
1318 75.01.24 rab dcl str auto char(length(char_star_fun(length(b))))); fails due to all_operators_nuc.
1317 75.01.24 eew (stringsize): bit2= fixoin; where fixoin has a large value, does not raise stringsize condition. Bug in assign_op.
1316 75.01.24 rhs bit(<expr> | <non_integer>) gives the wrong result.
1315 75.01.23 rhs var_char_arr | char_arr = char(var_bit_arr) produces code that fails
A 75.01.22 rhs storage condition raised in var_char_array | char_array | var_char caused by bad addressing for varying aggregate temporaries.
1313 75.01.21 mbw size condition not raised in decimal assignment when value too large.
1312 75.01.21 rhs fixeddec | ** | 2 gets wrong precision.
$1311\cdot 75.01.20$\ rab $\text{aggregate\_fun \ sim > \ aggregate\_fun, \ where \ both \ fun's \ are \ the \ same; fails.}$

$1310\cdot 75.01.20$\ rab $\text{bit\_vary \ fun} = x; \ where \ x \ is \ fixed \ bin \ with \ a \ scale \ greater \ than \ its \ precision, \ fails \ in \ that \ bit\_varying \ is \ given \ a \ negative \ length.自己$

$1309\cdot 75.01.15$\ rab $\text{get data fails for any based variable whose implicit qualifier is other than a simple reference to a scalar pointer variable (CS3I84). No room in runtime\_symbol node to encode a more complicated qualifier expression.}$

$1308\cdot 75.01.14$\ rab $\text{dcl a(10) init((5)(1,*)); gets FATAL ERROR 310.}$

$1307\cdot 75.01.13$\ rab $x(a) = y(b) \text{ ** a; can fail due to bad addressing code.}$

$1306\cdot 75.01.13$\ rab $\text{math builtins with decimal args should produce decimal results of the same precision.}$

$1305\cdot 75.01.13$\ rab $\text{abs(4)}, \text{complex(10,8)}, \text{etc. should produce decimal results.}$

$1304\cdot 75.01.13$\ rab $\text{get edit may fail with drifting sign picture format.}$

$1303\cdot 75.01.10$\ rab $\text{J}^b \text{ > } \text{M}^b \text{ fails.}$

$1302\cdot 75.01.10$\ rab $\text{translate(}a^a, \textABCDEF, \text{aaa}) \text{ }$gets $\text{D}$ instead of $\text{A}$.}

$1301\cdot 75.01.09$\ rab $\text{(any\_to\_any\_bug): } \text{-33000000000.0b} \text{ gets underflow while being converted to float binary.}$

$1300\cdot 75.01.08$\ rab $\text{bad relocation bits for internal file constants.}$

$1299\cdot 75.01.06$\ rab $\text{symbol table entry built incorrectly for var with position attribute.}$

$1298\cdot 75.01.08$\ rab $\text{tan(complex\_var) fails.}$

$1297\cdot 75.01.07$\ rab $\text{(any\_to\_any\_bug): float\_dec3 } = \text{8.23e-127}\text{; gets conversion instead of underflow.}$

$1296\cdot 75.01.07$\ rab $\text{onfile() not consistent about length of file name.}$

$1293\cdot 75.01.06$\ rab $\text{passing subst(str,1<expr>) as arg in a procedure with a large stack frame may fail.}$

$1292\cdot 75.01.06$\ rab $\text{call char\_star(before(char\_varying,"$	ext{3}"\text{")); gets ERROR 313. (builtin bug).}$

$1291\cdot 75.01.06$\ rab $\text{real\_decimal + complex\_decimal fails.}$

$1290\cdot 75.01.06$\ rab $\text{allocate stmt or allocation builtin get fault at runtime for internal controlled variable that is NOT set.}$

$1289\cdot 75.01.06$\ rab $\text{allocn not recognized as another name for allocation builtin.}$

$1288\cdot 75.01.06$\ rab $\text{dcl f file ext constant; gets ERROR 218.}$

$1287\cdot 75.01.06$\ rab $\text{string(valid(pot)) gets ERRORS 313 & 316 (builtin bug).}$

$1286\cdot 75.01.03$\ rab $\text{a(j) } = \text{b + c(j);} \text{ where a and b are fixed dec(7,3) and c is fixed bin tails (isn 3419).}$

$1285\cdot 75.01.03$\ rab $\text{put list(array\_fun()); puts only 1 element of array\_fun() (mpfr 7945).}$

$1284\cdot 75.01.03$\ rab $\text{call f(array); where f expects an array of different dimensionality than array does not get ERROR 79 as expected.}$

$1283\cdot 75.01.03$\ rab $\text{put data(structure\_star\_extents\_parameter); can fail.}$

$1282\cdot 74.12.31$\ rab $\text{put data(packed\_star\_array) and put list(packed\_picture\_array) can fail.}$

$1281\cdot 74.12.29$\ rab $\text{assignment of array cross section members of structures may fail (bug in expression\_semantics).}$

$1280\cdot 74.12.27$\ rab $\text{put list(convert(2.0D0eO+(2+3i)))); gets ERROR 253.}$

$1279\cdot 74.12.26$\ rab $\text{(optimizer bug): a=array(i+j);begin;j=j+1;end;b=array(i+j);}$

$1278\cdot 74.12.26$\ rab $\text{fails because optimizer fails to realize that stmts in begin block set inputs to expressions it has commoned.}$

$1277\cdot 74.12.21$\ rab $\text{string(define\_structure) gets FATAL ERROR 310 (mpfr 7909).}$

$1276\cdot 74.12.21$\ rab $\text{assignment of aggregate on character (non\_word) boundary so in test case. (C80457.)}$

$1275\cdot 74.12.21$\ rab $\text{call defined\_entry\_variable; gets FATAL ERROR 310.}$

$1274\cdot 74.12.13$\ rab $\text{d } = \text{6/2;} \text{ gets IPR fault if d is decimal number whose scale factor is larger than 32 or smaller than -31.}$
Conversion of fixed decimal to fixed binary by type instead of target can fail if source scale is outside range of -38 to +38.

Do i = 1 to 6/2; where i is fixed bin aets IPR fault.

Dcl a bit<const> int static init<const char_string_constant>; drops the last character of char_string_constant in initializing a.

Rhs (low priority) assignment of aggregates with refer extents does not quite work properly in that bounds checking should take place before any assignment. If checking is successful, a simple copy can be done.

Dcl p -> data(i,*); = q -> data(i,*); can fail when data has refer extents. (mprf 7865). Bug in expand_assign.

Rhs substr( picture_in_struct ,... fails. Built-in bug.

Rhs repeated use of substr(p(i)->varying,n) can cause FATAL ERROR 337 with optimize. U11 in outliner.

Rhs atan(dy,x) fails when dy is double precision and x is single. See test1263.

For atan(y,x) where y and x are float binary, the result precision should be the maximum of the precisions of y and x instead of the precision of y.

Dcl ... 3(10b) ... ; can get FATAL ERROR 336 with table option.

Rhs float_unal * float_unal can fail. (save_value bug.)

Rhs structure-valued function calls fail. (declare bug) (c52055).

Rhs dcl str auto char(length(char_star_func(length(b)))) where b has star extents gets FATAL ERROR 335. (mprf 7837)

Rhs subbstr(init<expr,72>,37) fails. Similar constructs also fail.

Rhs repeated use of dim and lbound ifs with arrays of star extents and optimize options can cause ERROR 313 (c80654) (prepare_operand bug).

Rhs imag(flt_cmp_bin(i)) fails -- bug in built-in.

Rhs unaligned member of auto structure passed incorrectly to quick proc if passed by ITP.

Rhs (all_operators_bug); return(char(*)); from inside a non-quick begin_block faults in all_operators_ (mprf 7815).

Rhs when a structure is erroneously declared without a level 1 component, the compiler faults after correctly printing out ERROR 16. (mprf 7812).

Rhs In rare circumstances, the use of an unaligned short bit string parameter in the statement immediately after it is set can cause ERR0K 313 with optimize.

Rhs In a multiple-entry proc with some params in different positions and a prologue, an incorrect entry sequence can be generated in rare circumstances.

Rhs while clause of do statement containing an irreducible varying or star extents function reference is compiled incorrectly.

If an array of substructures has refer extents for more than one bound, redundant assignment statements will be generated at allocate time to set up the bound bounds. Also, simplify_expression should be run on the word size of the struct if it is non-constant.

Aggregate statements involving members of arrays of structures with expression extents may fail if program has put data, get data statements or is compiled with table.

Use of nested initial-lists in initial attribute can cause compiler to loop. Bug in compile_tree for nested join ops.
1246 '74.11.31 gdc dcl str auto char[1:] in(func(0)) ; where func returns char(*) may get futal - 335. (mprf 7788)
1245 '74.11.01 rab enable_f operator sets spl5 to incorrect value.
1244 '74.10.26 rab onti: char sug: substr(strv(str)) = <expr>; substr(strv(str)) = expr; fails where f(str) is any expression with str as 0 or argument. Optimizer removes second statement. This is a rae
1243 '74.10.21 rab dcl 1 st based=2 at= array(m) float bin63); 2 bt=3 (y, z) fixed bin3 double float bin63); -y and doble get placed on odd rather than correct even bound. (mprf 7774)
1242 '74.10.19 rab Use or more than 1 short strings string_overlay defined on a short string member of a structure can cause bad code in various circumstances if optimizer used. (Not an optimizer bug. Cause to bug 941.)
1241 '74.10.19 rab varying_str_const(constant) passed by value to entry with char(*) or options(variable) causes ERROR 315.
1240 '74.10.14 rab execution of a complex binary comparison can cause zero divide condition. Hug in code generator.
1239 '74.10.07 rab complex float bin operations with constant operands can fail if the real part of a constant is an integer.
1238 '74.10.07 rab dcl foo int static init(<expr> or -<expr>); gets fatal process error instead of ERROR 324.
1237 '74.09.24 gdc index(substr(reverse(based_varying_str),i),comma) gets ERROR 313 (mprf 7686).
1236 '74.09.24 gdc imag(st.foo), where st is auto or static, can cause other rets to foo to be bad. bug in builtin.
1235 '74.09.23 rab fixedbin71 ** 1 gets ERROR 302.
1234 '74.09.23 rab fixedbin71 ** 0 fails (gets bad code and ERROR 302)
1232 '74.09.23 rab calls with more than 64 args cause ERROR 310. Should get implementation restriction message.
1231 '74.09.16 gdc hbound(array, var) gets fault or fatal process error at runtime (mprf 7662).
1229 '74.09.09 new substr(s,1,0) = ""; may fail in that the rest of s may be changed. Note that this does not fail when third arg is variable instead of constant zero.
1228 '74.09.09 rab (optimizer bug): if unal_str = "... then if unal_str = "... call foo(addr(unal_str)); sometimes fails because the addr of a temp created for unpacked value of unal_str is passed out. (mprf 7642).
1227 '74.09.09 rab return(<short_string_expr>) may fail with no optimize & many returns declarations (at entry pts). Remedy: use optimize.
1226 '74.08.29 rhs aggregate_fun1() <op> aggregate_fun2() fails -- bad code & ERROR 301.
1219 '74.08.10 rab bad code for more than one invocation of the same returns(char_or_bit(4)) function in the same statement.
1217 '74.08.09 rhs varying_ch_array, ch_array = char(bit_varying_array); gets ERROR 313 (mprf 7603).
1186 '74.07.23 rab dcl c char(<c) autonstatic unal;dcl cb char(<c) based(addr(c)); references to cb are not padded while references to c are. This is part of a more general problem which has always existed and will not be fixed quickly.
1185 '74.05.23 rab any_to any: hit to char conversion fails for strings longer than 1.
1178 '74.05.23 rab net data; fails if input variable is declared but not referenced in the program.
1177 '74.04.22 rab into from options and strings, unspec builtins fail for a connected array reference whose declaration doesn't have its own dimensions. This is a rare case. Example:
dcl 1 st(3), 2 a fixed bin; write from(a); Note that st
has no other members.

SUGGESTED IMPROVEMENTS
12 75.01.07 rhs onloc() does not recognize quick procedures.
12 74.08.27 rhs x(i,*) = u(i,*) where x is based produces a temporary of the
wrong size, possibly causing FATAL ERROR 308.
1110 74.05.16 rhs %include statement on line 2 causes strange listing (mprf 7190).
1033 74.02.26 rhs If the 2nd or 3rd argument of translate bit or 2nd argument of i
or verify bit is constant, the allocate bit in the symbol node is
be turned off so that the constant is only allocated if it is re
needed (mprf 6815).
777 73.12.28 rhs string(char_array(i, *)) gets ERROR 272 (mprf 6367).
697 73.12.27 rhs compiler should diagnose use of illegal exprs in init attribute of
static vars (mprf 6154).
24 73.12.26 rhs dcl nname fixed bin; a %include statement; where statement has a
complete declaration gets misleading error message in that "a" is
not printed (mprf 5883).
597 73.12.13 rhs LEX: comments and blank lines should be allowed after logical er
46 73.12.05 rhs symbol map listing should include file attributes, e.g., input, c
142 73.09.02 rhs dcl A(n,m) float bin based(p), B(n,m) float bin based(q);
A(i,*) = B(i,*)
generates assignment to temporary when
A(i,*) = B(i,*)
uses copy words operator directly into A. Furthermore, size of temporary used is wrong! Instead of allocating space for
one row (m words), space for entire array (n*m) words is
allocated!
87 72.10.06 blw put transfer vectors in listing (mprf 4877)
78 72.11.03 rhs suppress conversion message for refer(struct_bit) when substr pro
81 72.08.10 rhs (subrg); array = scalar; / can suppress range checking code
28 72.07.10 rhs remove extra unneeded assignment in subscript expr involving fix
20 72.07.03 rhs make two entries in profile for "if ... then ...",
TITLE: Fix reused address bug in Salvager

AUTHOR: A. Kobziar

SUMMARY:
Salvager tries to handle a reused address in a 256k segment with a 64k page table. This results in a subsequent crash. Fix the zeroing of reused addresses by putting only the address of the one page into a small page table rather than all 256 addresses.

DETAILED PROPOSAL:
Coded, tested, and sent to Phoenix as an emergency fix.
**Multics Change Request**

**TITLE:** Install compare_mst

**AUTHOR:** A. Kobziar

<table>
<thead>
<tr>
<th>Code</th>
<th>Category (Check One)</th>
<th>Document (Specify One or More)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔ PL/I</td>
<td>Lib. Maint. Tools</td>
<td>DOCUMENTATION CHANGES</td>
</tr>
<tr>
<td>✔ ALM</td>
<td>Sys. Maint. Tools</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Status**: A 04/08/75
- **Expires**: 10/08/75

- **Explain in DETAILED PROPOSAL**
- **Planned for System MR**
- **Fixes Bug Number(s)**

- **Documented in MBT**
  - 355

- **User/Operations-visible Interface change?**
  - yes  x  no

- **Incompatible change?**
  - x yes  x no

- **Performance:**
  - x Better  x Same

- **Replaces MCR**

**Objections/Comments:**

Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)

**SUMMARY:**

Install compare_mst in the Tools library. This command prints a summary of differences between two Multics System Tapes. (service, salvager, or bos).

**REASONS:**

A standard Multics release ships source, object, and a new system tape. By generating a new system from this object and comparing it to the released system, one can determine if the correct object has been shipped. Comparing MST tapes also provides an easy method for checking for undetected tape errors in the object. For development machine users, it provides a method for double checking the completeness of a hardcore submission.

This command should also be useful when one has an MST, but is not sure exactly what is on it. By comparing the unknown MST to a known MST, all differences will be listed.

**DETAILED PROPOSAL:**

See attached draft documentation.
This command is used to read two MST tapes and to list all differences between them. All differences in segment headers, and the starting address of any inequalities or differing lengths of segment contents are noted. Additions, deletions, and moves of segments are handled. One can optionally save the contents of differing segments in the user's working directory for further detailed comparisons. Any number of collections can be handled, but a warning message is printed if a tape does not end in a collection mark.

Usage:

```plaintext
compare_mst tape1 tape2 -control_arg
```

tape1 is the tape label of the old tape
tape2 is the tape label of the new tape
control_arg:

- `save` if any discrepancies in the contents of corresponding segments are noted, the contents will be saved in the user's working directory under the names "tpl.<segment_name>" and "tp2.<segment_name>". An added segment will be saved under the name "tp2.<segment_name>".
SUMMARY:

1. Fix unreported and reported bugs.
2. Improve data initialization code.
3. Implement new interface with code generator.
4. Make use of improved optimizer and code generator.

REASONS:

1. Initial implementation of data initialization required 4 or more external calls per array element. New implementation requires none.
2. Code generator was not informed that the optimizer was invoked, although that information is useful to it.
3. The soon to be installed PL/I compiler generates improved code for many constructs.
This file contains fortran BUGS, SUGGESTED MOVEMENTS, PLANNED LANGUAGE FEATURES, UNDIAGNOSED PROBLEMS, and FURTHER INFORMATION.

BUGS marked with * have been fixed in the experimental compiler located in $ldde$.
BUGS without * will generally exist in both the installed compiler and the experimental compiler.

147 next free number
141-75.03.24 dsl Unformatted I/O (both random and sequential), does not work at all. This is due to an unset variable in the I/O package.

140-75.03.21 dsl If a member of common appears in a namelist group, the common block must be allocated or namelist I/O using that group will fail. This only affects common blocks which are only referenced in namelist groups.

139-75.03.19 dsl Input errors involving namelist and list-directed input will probably result in fatal process errors.

138-75.03.10 rab Incorrect subscript code is generated for arrays whose starting element is more than 64k words from the beginning of a segment. Use of the optimizer may alleviate the problem.

137-75.03.10 rab Optimizer bug for arithmetic if statements. If any target results in a backward transfer, then subsequent targets which are forward transfers will not be optimized.

136-75.03.10 dsl A bug in the code generator prevents debug and probe from obtaining valid runtime addresses for members of FORTRAN common.

135-75.03.10 dsl The two character sequence "$F", where F is uppercase, should be recognized as an end of file indicator.

134-75.02.26 dsl When processing carriage control characters on output, no attention is payed to the hardware restrictions concerning overlapping fields. Very rarely, this results in garbled output. The occurrences are infrequent and many times not reproducible.

133-75.02.26 dsl Mode statements of the form:
   mode*k_h_name ...
where "$h_name" is any name starting with a lower case "$h", will not compile correctly. This is due to the compiler interpreting decimal integer followed by "$h" as a hollerith constant.

132-75.01.21 dsl If a symbol has both the external and parameter attributes, it is an entry variable not an entry constant. The link should be to the value of the variable. For example-

   subroutine test(x,y)
   external x
   y = x(y)
   should generate a link to the entry variable x, not the entry constant x.

131-74.12.05 dsl If a character array appears in a call statement, then I/O statements referencing that array name may be compiled incorrectly. Scalar references are not affected.

130-74.12.05 rab Code generator does not handle arithmetic if statement when the three target labels are the same.

129-74.11.22 dsl If the -card or -convert option is given, the source pathname, unique id, and date-time-modified stored in the object segment will be incorrect.

128-74.11.22 dsl If an internal builtin function name is declared with a mode
which is not the same as the mode of that builtin function, the name should not acquire the builtin attribute.

If an internal builtin function name is declared with the external attribute, it should not acquire the builtin attribute.

Incorrect code is generated for alternate return statements if the expression is not integer.

Character assignment statements whose right hand side is a character function are not compiled correctly. The result of the function should be placed in a temporary if the character length of the function is not equal to the length of the left hand side.

The correct argument descriptor is not being generated for elements of parameter variable bound arrays. If an element of a parameter variable bound array appears in an external call or function reference, the descriptor for the entire array is currently used.

The entry point code does not pick up a pointer to the parameter descriptors. This will prevent access to these descriptors. This only affects references to parameter array descriptors and will only affect call statements and function references which reference the entire array. (See also bug 124.)

Certain subscript expressions involving constants result in a fatal process error during compilation. The form affected is constant - simple_reference.

Complex builtins and relational expressions with constant arguments can fail to compile correctly if the real part of any constant is an integer. This is identical to PL/I bug 1239.

A fatal process error may occur instead of an error message for list-directed input errors.

Alternate return labels are not implemented correctly. Attempts to use alternate return labels will generally result in a fault.

Incorrect code is generated for subscripted references when the -subscriptrange (-subrg) options is specified. Code is correct if this option is not used.

Variables declared by parameter statements were implemented incompletely. As a result these variables could not appear in parameter lists.

SUGGESTED IMPROVEMENTS

Incluce the bit representation of the offending character in the error message for error 75.

Implied do loops in data statements are costly. Their usage should be avoided where the feature is unnecessary. (e.g. -dimension a(10,10)/data ((a(i,j)), i = 1, 10), j = 1, 10) IS EXPENSIVE)

The following improper sequence is not diagnosed:

```
do 10 i = 1, 10
   ... if (....) go to 10
   do 10 j = 1, 10
   ...```

Type conversion of constants is sometimes delayed until execution, resulting is less efficient code. For example:
dc 100 x = 0, 13.
If $x$ is a real variable, the constant 0 will be converted
to floating point (0.0) everytime the statement is
executed.

142 75.03.25 dsl Mode statements of the form:
        `mode *k
are emitted by the compiler when using the -convert option,
although they are not acceptable to the compiler without
this option.

PLANNED LANGUAGE FEATURES

FURTHER INFORMATION

When a problem is discovered, it is given a date and a serial number.
An attempt is made to produce a small test case which exhibits the problem.
Where it is helpful, the relevant parts of the test case will appear with
the description of the problem.

Some test cases are saved and used to check for compiler regression.

If it appears the compiler is "looping", QUIT, then type trace_stack.
Submit the trace_stack output and a description of your problem through
the standard bug reporting procedure of your installation.

(END)
Request 04/10/75 1605.0 mst Thu
Output 04/10/75 1607.7 mst Thu
Print queue 3

146 lines, 5 pages
85 blocks at $0.50 per 1000 blocks
Total charge to RBarnes.Multics.m
TITLE: Delete Version 1 Graphics Software

AUTHOR: John Gintell

SUMMARY: Delete Version 1 Graphics software from the system.

REASONS:

Version 1 Graphics software is incompatible with Version 2 which has recently been approved for addition to the system.

It is badly implemented, unnecessarily complicating the ring0 Typewriter DIM.

DETAILED PROPOSAL:

1. Remove graphics handling from ring0 Typewriter DIM (MCS).

2. Delete bound_graphics_.

3. Delete bound_lsm_.

4. Remove lsm and graphics error codes from error table.

IMPLICATIONS:

Users of Version 1 graphics have already been warned that it will be replaced.
**MULTICS CHANGE REQUEST**

**TITLE:** New date/time strings for `convert_date_to_binary_`

**AUTHOR:** Gary C. Dixon

**Planned for System:** not applicable

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

**Documented in MTB:** 155

**Incompatible Changes:** no

**User/Operations-visible Interface Change:** yes

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

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

**DOCUMENTATION CHANGES (specify one or more):**
- **MPM (vol,sect)** MPM Sub. MPAM (sect)
- **MPAM (sect)**
- **MOSN (sect)** MSAM (sect)
- **PLMs (AN#)**
- **Info Segs**
- **Other** MPM Subroutines is AG-93

**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:**
Make an upwards compatible extension to the date/time strings accepted by `convert_date_to_binary_` to simplify the format of the string and make it more natural to use.

**REASONS:**
People at the Design Review for MTB-155 concluded that it was extremely awkward and unnatural to use the offset feature (eg, 1 month, 5 days, Saturday) in the current date/time string format, and that the extensions proposed for offsets in MTB-155 would further complicate the format and would lead to some ambiguities in the meaning of date/time strings (eg, 03/01/75 + 1 day +1 month). One of the suggestions resulting from MTB-155 was extended to cover all cases, and is shown in the DETAILED PROPOSAL below.

**IMPLICATIONS:**
Users should find date/time strings containing offsets more natural to use and less ambiguous in their meaning. Backward offsets can be added to the date/time string in a more natural way.

**DETAILED PROPOSAL:**
1) New Features 3 through 11 and all of the Changed Features were well-accepted by the Design Review, and would be incorporated in the change. These features were deemed beneficial, no matter what date/time format was used.
2) Date/time strings, as accepted today, would continue to be accepted, with the exception of the three items mentioned in Changed Features section of MTB-155. It was deemed that these changes would not affect users, but instead correct bugs in the current version of convert_date_to_binary.

3) Install encode_clock_value_ and decode_clock_value_, as documented in MTB-155.

4) Change the documented specification for the date/time string to be as follows:

```
  OFFSET... [CONSTRAINT]  [today] [now] [LOCAL-ZONE]
  DATE          TIME     TIME-ZONE
```

The new date/time string would be processed from right to left, first interpreting the absolute date and time, then applying any offsets which are given in their order of appearance.

The DATE would be specified as it is today. New Feature 5 of MTB-155 would add a new date format:

```
YEAR_OF_CENTURY MONTH DAY
```

For example, 75.2.1 for February 1, 1975. In addition, because some date specification would be required when a CONSTRAINT or OFFSET was given, the new term "today" would be allowed, representing today's date which is the default if no DATE is given.

As indicated above, the CONSTRAINT can only appear if a DATE is given. The CONSTRAINT constrains that the specified date falls on the given day of the week. The format of a DATE-CONSTRAINT phrase would be:

```
DATE falls on DAY_NAME
```

where DAY_NAME is the name of a day of the week. If DATE did not fall on the specified DAY_NAME, an error would occur.

The TIME would be specified as it is today. New Feature 6 of MTB-155 would allow extra digits to be used in the fractional minutes field of time format:

```
HHMM.MMMMMHMM
```

where HH is an hours specification, MM is a minutes specification, and MMMMMHMM is an optional fraction of minutes. If no TIME is given, the default time is the current time. A new term "now" would be allowed, so that this default could be explicitly specified when used with a time OFFSET.

The TIME-ZONE would be specified as it is today. Different time zones could be defined at individual installations by changing the values in the time_table_ data base, proposed as New Feature 10 in MTB-155. If no TIME-ZONE were specified, the local time zone (from sys_info) would be
As mentioned above, a DATE and/or TIME would be required whenever an OFFSET was used. The OFFSETS would have to precede the DATE and TIME, and would be applied to the DATE/TIME in their order of appearance from right to left. Two forms of offsets would be allowed: day name offsets; and time period offsets. These forms are shown below.

### DAY NAME OFFSET

<table>
<thead>
<tr>
<th>Offset Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY_NAME before DATE/TIME</td>
<td>Tuesday before 04/07/75</td>
</tr>
<tr>
<td>DAY_NAME on or before DATE/TIME</td>
<td>Monday on or before 04/07/75</td>
</tr>
<tr>
<td>DAY_NAME before or on DATE/TIME</td>
<td>Monday before or on 04/07/75</td>
</tr>
<tr>
<td>DAY_NAME after or on DATE/TIME</td>
<td>Monday after or on 04/07/75</td>
</tr>
<tr>
<td>DAY_NAME on or after DATE/TIME</td>
<td>Monday on or after 04/07/75</td>
</tr>
<tr>
<td>DAY_NAME after DATE/TIME</td>
<td>Tuesday after today</td>
</tr>
</tbody>
</table>

where DAY_NAME is the name of any day of the week.

### TIME PERIOD OFFSET

<table>
<thead>
<tr>
<th>Offset Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER PERIOD before DATE/TIME</td>
<td>2 weeks before 04/07/75</td>
</tr>
<tr>
<td>NUMBER PERIOD after DATE/TIME</td>
<td>3 days after today</td>
</tr>
<tr>
<td></td>
<td>10 minutes after now</td>
</tr>
</tbody>
</table>

where NUMBER is a non-negative integer, and PERIOD is one of the time periods shown in the table below. If the NUMBER were 1, a singular PERIOD would be required. Otherwise, a plural PERIOD would be required. PERIOD abbreviations would be treated as both singular and plural.

<table>
<thead>
<tr>
<th>PERIOD</th>
<th>PERIOD Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>second(s)</td>
<td>sec</td>
</tr>
<tr>
<td>minute(s)</td>
<td>min</td>
</tr>
<tr>
<td>hour(s)</td>
<td>hr</td>
</tr>
<tr>
<td>day(s)</td>
<td>da</td>
</tr>
<tr>
<td>week(s)</td>
<td>wk</td>
</tr>
<tr>
<td>month(s)</td>
<td>mo</td>
</tr>
<tr>
<td>year(s)</td>
<td>yr</td>
</tr>
</tbody>
</table>

**EXAMPLES:**

Make sure specified date falls on Tuesday:

```
    memo -time "4/8/75 falls on Tuesday" Meet on Tuesday
```

Invoke crank absentee job on last day of this month:

```
    ear crank -time [date_time 1 day before 1 month after [month]]/1 midnight
```
**SUMMARY:**

Change the `get_quota` command to call `hcs_$quota_read` instead of the obsolete entry `hcs_$quota_get`.

**REASON:**

The new entry returns a fixed `bin(71)` value for the time-page product instead of a fixed `bin(35)` value and the `gq` command was never updated to handle the larger value.
**Title:** Fix bug in initiate command  

**Author:** S. Herbst

- Coded in [X] PL/I [ ] AIM [ ] other- 
- explain in DETAIL PROPOSAL  
- Planned for System MR  
- Fixes Bug Number(s)  
- Documented in MTC  
- User/Operations-visible  
- Interface change? [X] yes [ ] no  
- Incompatible change? [ ] yes [X] no  
- Performance: [ ] Better [X] Same  
- Replaces MCR  

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

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

**Objections/Comments:**  
- None (Reason) NO CHANGE

**Use these headings:** SUMMARY, REASONS, IMPLICATIONS, DETAIL PROPOSAL (Optional)

**SUMMARY:**

Fix bug in the initiate command whereby bad control arguments are interpreted as reference names.

**REASON:**

Currently, any argument beginning with a minus sign other than -s, -lg, and -long is made a reference name on the segment being initiated.