

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 01 August 75 through 15 August 75.

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

TITLE: Make runoff_abs take dprint control args		STATUS	DATE	
AUTHOR: S. Herbst		Written	07.07.75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)	Status	01/15/75 18/5/75	
	<input type="checkbox"/> Lib. Maint. Tools	Expires	2/5/76	
	<input type="checkbox"/> Sys. Anal. Tools	DOCUMENTATION CHANGES		
	<input type="checkbox"/> Sys. Prog. Tools	Document	Specify One or More	
	<input type="checkbox"/> 355	MPM (Vol, Sect.)	II, 1	
	<input type="checkbox"/> BOS	PLMS (AN #)		
	<input type="checkbox"/> Salvager	MOSN (Sect.)		
	<input type="checkbox"/> Ring Zero	MPAM (Sect.)		
	<input type="checkbox"/> Ring One	MSAM (Sect.)		
	<input type="checkbox"/> SysDaemon/Admin.	Info Segs		
<input type="checkbox"/> Runtime	Other (Name)			
<input checked="" type="checkbox"/> User Cmnd/Subr.	None (Reason)			
Objections/Comments:				
Use MPM write up (attached) as info segment				

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

SUMMARY:

Make runoff_abs accept all the dprint control arguments, as was done for the translator abs commands.

MFM commands

Name: runoff_abs, rfa

This command submits an absentee request to process text segments using the runoff command. The absentee process prepares, in manuscript form, an output segment for each text segment and stores each output segment in the user's working directory. The name of the output segment is the name of the text segment with the suffix runoff replaced by the suffix runout. The absentee process then uses the dprint command to queue each output segment for printing and deletion. Printing or deletion can be withheld if desired. If the -output_file control argument (one of those recognized by the enter_abs_request command) is not specified, the absentee process' output segment is placed in the user's working directory with the name path1.absout, where path1 is the first argument of the command. (See "Usage" below.)

Usage

```
runoff_abs paths -rf_args- -ear_args-  
                -dp_args- -control_args-
```

where:

1. paths are the pathnames of segments to be processed by the runoff command. If the suffix runoff is not present, it is assumed. If more than one pathname is given, each segment is considered a separate runoff task.
2. rf_args can be one or more control arguments accepted by the runoff command.
3. ear_args can be one or more control arguments accepted by the enter_abs_request command with the exception of the -brief (-bf) control argument.
4. dp_args can be one or more control arguments accepted by the dprint command.
5. control_args can be:
 - hold specifies that the output segments created by runoff should neither be queued for printing nor deleted. Each output segment is formatted for printing on a selectric-type terminal, with a 963

help -pn runoff_abs
(21 lines follow)
03/15/74

Name: runoff_abs

The command runoff_abs (rfa) submits an absentee request to runoff segments. The absentee process for which runoff_abs submits a request will prepare each segment named in manuscript form, and store each output segment in the user's current working directory. The name of the output segment is the name of the segment plus the suffix ".runout". runoff_abs then queues each output segment for printing and deletion by the i/o daemon. Printing and deletion can be withheld if output through another device is desired. If the -output_file control argument is not specified, the output segment of the absentee process will be placed in the user's current working directory, and its name is the name of the segment plus the suffix ".absout". If more than one segment is specified the first one will be used.

Usage

56 lines follow. More help? yes

```
runoff_abs path1 ... pathn -runoff_control_arguments-  
-enter_abs_request_control_arguments-  
-dprint_control_arguments-  
-runoff_abs_control_arguments-
```

1. path *i* can be absolute or relative pathnames. Specifies the segment to be processed by the runoff command. If more than one pathname is given, each segment will be considered a separate runoff task.
2. runoff_control_arguments: may be one or more control arguments accepted by the runoff command. See its description in MPM or type "help runoff". Control arguments must begin with "-".
3. enter_abs_request_control_arguments: may be one or more control arguments accepted by the enter_abs_request command, except "-bf" or "brief". See its description in the MPM or type "help enter_abs_request". Control arguments must begin with "-".
4. dprint_control_arguments: may be one or more control arguments accepted by the dprint command. See its description in the MPM or type "help dprint". Control arguments must begin with "-".
5. runoff_abs_control_arguments:
-copy *n*, -op *n* specifies the number of copies of the segment to be dprinted ($1 \leq n < 4$); The default is 1.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1311
TITLE: Fix bug in >tools>If		STATUS: DATE
AUTHOR: Txom McGary		Written: 75.07.23
		Status: A 08/05/76
		Expires: 01/23/76
Planned for System: not applicable		CATEGORY (check one)
Fixes Bug Number(s): not applicable		() Lib. Maint. Tools
Documented in MTB: not applicable		() Sys. Anal. Tools
Incompatible Change: no		(<input checked="" type="checkbox"/>) Sys. Prog. Tools
User/Operations-visible Interface Change: no		() 355
Coded in: (<input checked="" type="checkbox"/>) PL/I () ALM () other-see below		() BOS
Performance: (<input checked="" type="checkbox"/>) better () same () worse		() Salvager
DOCUMENTATION CHANGES (specify one or more)		() Ring Zero
MPM (vol,sect) MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		() SysDaemon/Admin
PLMs (AN#) AN51 (System Tools)		() Runtime
Info Segs		() User Command/Subr
Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY: For the cases
 if less N N -then whatever
 and
 if greater N N -then whatever
 if complains:
 if: unknown keyword "greater", no execution

REASON: This is due to missing "else" clause for case of two equal arguments. (Control falls into the code for unrecognised keywords.)

IMPLICATIONS: Documented feature will now work.

TITLE: Fix bug in mseg_index_		STATUS	DATE
AUTHOR: Jerry Stern		Written	7/25/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR <u>3.0</u> -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 08/05/75
		Expires	02/05/76
		DOCUMENTATION CHANGES	
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input checked="" type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input type="checkbox"/> Ring Zero <input checked="" type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input type="checkbox"/> User Cmnd/Subr.		Document	Specify One or More
Objections/Comments:		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason)	<input checked="" type="checkbox"/>

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

Summary: Fix a bug in the program mseg_index_ that allows a process to open a message segment to which it does not have "rw" effective access.

Reasons: Opening a message segment without "rw" effective access is meaningless since all message segment operations require it. Although message segment ACLs are always set to "rw", the Access Isolation Mechanism removes "w" effective access in certain cases. This would result in a no-write-permission fault and a crawl-out from ring 1.

Implications: None

Multics Change Request

MCR 1313
Page 1 of 2

TITLE: Install - init_ftn :_builtin
AUTHOR: D. S. Levin

STATUS	DATE
Written	07.24.75
Status	A 08/05/75
Expires	02/05/76

- Coded in PL/I ALM other-
explain in DETAILED PROPOSAL
- Planned for System MR
- Fixes Bug Number(s)
- Documented in MFB
- User/Operations-visible
- Interface change? yes no
- Incompatible change? yes no
- Performance: Better Same
 Worse
- Replaces MCR

Category (Check One)	
<input type="checkbox"/>	Lib. Maint. Tools
<input type="checkbox"/>	Sys. Anal. Tools
<input checked="" type="checkbox"/>	Sys. Prog. Tools
<input type="checkbox"/>	355
<input type="checkbox"/>	BOS
<input type="checkbox"/>	Salvager
<input type="checkbox"/>	Ring Zero
<input type="checkbox"/>	Ring One
<input type="checkbox"/>	SysDaemon/Admin.
<input type="checkbox"/>	Runtime
<input type="checkbox"/>	User Cmd/Subr.

DOCUMENTATION CHANGES

Document	Specify One or More
----------	---------------------

MPM (Vol, Sect.)	
PLMS (AN #)	54
MOSN (Sect.)	
MPAM (Sect.)	
MSAM (Sect.)	

Objections/Comments:
Detailed information will be included in the fortran PLM.

Info Segs
Other (Name)
None (Reason)

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

SUMMARY:

Install init_ftn_builtin, a tool which is necessary to create the FORTRAN compiler.

REASONS:

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

DETAILED PROPOSAL:

init_ftn_builtin is used to initialize fortran_data a data segment that is a component of the FORTRAN compiler used in processing builtins.

Name: init_ftn_builtin

The execution of this program initializes an array of n structures in `fortran_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: init_ftn_builtin

To initialize the `builtin_name` array in `fortran_data` the following steps are performed:

- 1) extract the source program `fortran_data.alm`.
- 2) assemble `fortran_data.alm` in the working directory in which `init_ftn_builtin` is to be executed.
- 3) execute `init_ftn_builtin`.
- 4) place the newly-created `fortran_data` object segment in its archive.

TITLE: Fix bug in delete_		STATUS	DATE
AUTHOR: S. Herbst		Written	07.28.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 08/05/75
		Expires	02/05/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	no change

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

SUMMARY:

Fix bug in delete_ that sometimes causes an infinite loop.

REASON:

Loop that forces access assumes that a segment to be deleted can be initiated once the caller has been placed on its ACL with rw access. In the AIM context, it is possible for a user to have rw on the ACL of a segment and still not be able to initiate the segment.

DETAILED PROPOSAL:

Force access at most once.

Multics Change Request

TITLE: Fix move to respect safety switch		STATUS	DATE
AUTHOR: S. Herbst		Written	07.29.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 08/05/75
		Expires	02/05/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		Category (Check One)	
		Lib. Maint. Tools	
		Sys. Anal. Tools	
		Sys. Prog. Tools	
		355	
		BOS	
		Salvager	MPM (Vol, Sect.)
		Ring Zero	PLMS (AN #)
		Ring One	
		SysDaemon/Admin.	MOSN (Sect.)
		Runtime	MPAM (Sect.)
		X User Cmnd/Subr.	MSAM (Sect.)
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	no change

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

SUMMARY:

Fix the move command to query the user before deleting the source segment if its safety switch is on.

REASONS:

Currently, deletion of the source segment is forced and the meaning of the safety switch is disregarded.

Name: move, mv

The move command causes a designated segment or multisegment file (and its access control list (ACL) and all names on the designated file) to be moved to a new position in the storage system hierarchy.

Usage

move path1_i path2₁ ... path1_n -path2_n- -control_arg-

where:

1. path1_i is the pathname of the segment or multisegment file to be moved.
2. path2_i is the pathname to which path1_i is to be moved. If the last path2 segment is not given, path1_n is moved to the working directory, and is given the entryname path1_n.
3. control_arg can be -brief or -bf. This argument causes the messages "Bit count inconsistent with current length..." and "Current length is not the same as records used..." to be suppressed.

Notes

The star and equal conventions can be used.

When an entry is moved, it is given all of the names that the path1_i argument already has plus the entryname specified in the path2_i argument.

Since two entries in a directory cannot have the same entryname, special action is taken by this command if the creation of a segment or multisegment file would introduce a duplication of names within the directory. If an entry with the entryname path2_i already exists in the target directory and this entry has an alternate name, the conflicting name is removed and the user is informed of this action; the move then takes place. If the entry having the entryname path2_i has only one name, the entry must be deleted in order to remove the name. The user is asked if the deletion should be done; if the user answers "no", the move does not take place.

Read access is required for path1_i. Status and modify permissions are required for the directory containing path1_i. Status, modify, and append permissions are required for the directory containing path2_i.

The initial ACL of the target directory has no effect on the ACL of the segment or multisegment file after it has been moved. The ACL remains exactly as it was in the original directory.

If path1_i is protected, the user is asked whether to delete it.

TITLE: Modifications to probe		STATUS	DATE
AUTHOR: J. Broughton		Written	07.24.75
<input checked="" type="checkbox"/> Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL <input type="checkbox"/> Planned for System MR <input type="checkbox"/> Fixes Bug Number(s) various <input type="checkbox"/> Documented in MFB <input type="checkbox"/> User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse <input type="checkbox"/> Replaces MCR		Status	A 08/05/75
		Expires	02/05/76
		DOCUMENTATION CHANGES	
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input checked="" type="checkbox"/> User Cmmnd/Subr.		Document	Specify One or More
Objections/Comments:		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason)	no change

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

SUMMARY:

Make various modifications, bug fixes and improvements, to probe:

Bugs:

1. "go <nl>" is incorrect syntax and is not checked for.
2. "value octal (fixed bin35)" displays a double word in octal as opposed to the one word containing the variable.
3. A fault occurs in certain cases when a non-existent source line is referenced.
4. Label variables are not handled properly in some cases when used as statement designators, i.e. "position label_variable".
5. After resetting the last break in a segment, a fault will occur if the program is halted by the break and a "continue" request is issued which currently causes probe to return into the (non existent) break map.
7. A break set on a xec of an EIS instruction in PL/I operators does not work because the effective instruction length (including the descriptors following the xec) is not calculated correctly. (MPRf 8170)

8. Use of unary minus, e.g., "value array (-1)", causes fault. (MPRF 8191)
9. Handle fortran arrays correctly. Currently subscripts must be entered in reverse order.
10. Use of a cross-sectioned array reference as an operand to probe builtin is not checked for and causes fault - e.g., "value substr (array (1:2), 1,2)" (MPRF 8264)
11. If no locator is specified for a based array cross-section, a fault will result.
12. substr (string, any, 0) fails to give correct result, i.e., "". (MPRF 8403)
13. Handle bound segments with non-standard formats. The documentation implies that probe may be used with any program compiled by PL/I or FORTRAN; this change will allow it to work with standard format components even if the bound segment is non-standard.

Improvements:

1. Check for null environment pointer in label used in goto request to avoid fault and give user a better idea of what is happening.
2. Print label/entry variables as pointer pairs.
3. Handle picture variables in assignment, displaying their value, and giving their attributes.

REASONS:

To conform to documentation and to make probe a more useful tool.

DETAILED PROPOSAL:

Changes coded and have been tested as a private copy used by various people for about a month.

TITLE: Change probe to use new object map format		STATUS	DATE
AUTHOR: J. Broughton		Written	07.24.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	fl 08/05/75
		Expires	02/05/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	no change

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

SUMMARY:

Change probe to recognize both old and new format object maps and set the correct break map fields accordingly.

REASONS:

To support separate static, new fields have been added to the object map changing the location of the fields describing the break map. Therefore probe must check the version number to select the structure to be used.

DETAILED PROPOSAL:

Change break_manager_ to perform check.

TITLE: Installation of firmware utilities		STATUS	DATE
AUTHOR: L. Johnson		Written	07.29.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		
	<input checked="" type="checkbox"/> Lib. Maint. Tools	Status	A 08/05/75
	<input type="checkbox"/> Sys. Anal. Tools	Expires	02/05/76
	<input type="checkbox"/> Sys. Prog. Tools	DOCUMENTATION CHANGES	
	<input type="checkbox"/> 355	Document	Specify One or More
<input type="checkbox"/> BOS	MPM (Vol, Sect.)		
<input type="checkbox"/> Salvager	PLMS (AN #)	Tools	
<input type="checkbox"/> Ring Zero	MOSN (Sect.)	X	
<input type="checkbox"/> Ring One	MPAM (Sect.)		
<input type="checkbox"/> SysDaemon/Admin.	MSAM (Sect.)		
<input type="checkbox"/> Runtime	Info Segs		
<input type="checkbox"/> User Cmd/Subr.	Other (Name)		
	None (Reason)		

Objections/Comments:

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

SUMMARY:

Firmware modules for MPC's should be kept in a standardized place in the hierarchy. The suggested location is
 >ldd>firmware>firmware.archive.

REASONS:

Firmware programs are required by T&D and should be kept online.

DETAILED PROPOSAL:

1. Install a command load_firmware_file (lff) which creates segments from the firmware tape. These segments would then be combined into firmware.archive using the archive command. An archive is used to conserve space, since the firmware includes 500 small test program about 500 words long.
2. Install a procedure firmware_util_ for finding a firmware program. This procedure would check the working directory and >ldd>firmware for the archive and use archive_util_ to locate the module required. The procedure would return a pointer to, and the length of the firmware segment.
3. See the attached descriptions.

load_firmware_file

load_firmware_file

Name: load_firmware_file, lff

The load_firmware_file command loads MPC firmware modules from a firmware tape into segments. The command also generates a printable ASCII segment which contains a directory of those modules found on the tape.

Usage: load_firmware_file tape_name -control_args-

1. tape_name the name of the firmware tape to be used.
2. control_args optional control arguments which determine which modules are to be selected from the tape, and where they are to be stored. The control arguments may be selected from the following list:
 - pathname path specifies the name of the directory in which the firmware segments are to be stored. If omitted, they are stored in the working directory.
 - pn path
 - config specifies that the configuration deck is to be scanned to determine which devices are present in the system that require firmware.
 - device dev₁ .. dev_n specifies a list of up to 32 devices for which firmware programs are to be selected.
 - dv dev₁ .. dev_n
 - name name₁ .. name_n specifies a list of up to 32 program names which are to be loaded.
 - nm name₁ .. name_n
 - file file₁ .. file_n specifies a list of up to 32 file numbers which are to be included from the tape.
 - appl specifies that application firmware modules are to be loaded.
 - mdr specifies that micro-coded device routines (mdr's) are to be loaded.
 - itr specifies that integrated test routines (itr's) are to be loaded.

load_firmware_file

load_firmware_file

-list, -ls

specifies that no firmware segments will be created; only a listing of the contents of the tape will be generated.

-segment, -sm

specifies that the first argument (tape_name) is the pathname of a syspunch segment produced by the gc05 simulator containing the firmware object modules to process. No tape is required if this control argument is used.

Notes

The default action of this command, if no control arguments are specified, is to create a firmware segment from every unique module found on the firmware tape. This will require approximately 700 Multics records of storage.

If control arguments are specified to select which modules are to be loaded, only modules which match all of the criteria given will be loaded. For example:

```
lff firmware -file 3 -name btlt
```

will only load the program "btlt" if found in file 3, regardless of what other files may contain a "btlt" program.

If neither -appl, -mdr, or -itr control arguments are specified, the program will assume all types are to be loaded.

If the -config control argument is used, -device, -file, or -name may not be specified.

Use of the -device, -name, and -file control arguments requires some knowledge of the firmware tape. A listing of the contents of the firmware tape should be examined before using these arguments. The -device control argument is matched against columns 43-48 of the \$object card when scanning the tape. Typical devices are mtc500, urc002, ndm450, etc. The -name control argument is matched against columns 73-76 of the \$object card, which corresponds to the data on the LABEL card when the program is assembled.

load_firmware_file

load_firmware_file

Firmware segments are created using a 3 component entry name, in the following format:

fw.name.ident

where "name" is the name of the program and "ident" is taken from the MPC-assembler IDENT pseudo-op card.

A listing segment is always created which contains a directory of the firmware modules loaded. This segment has the name "tape_name.list".

Firmware modules are normally kept in the segment >ldd>firmware>firmware.archive. This archive should be created or updated after running load_firmware_file. The following is a typical example:

```
lff firmware -config
scl 16*modules loaded)
ac ad >ldd>firmware>firmware.archive [segs fw.**]
```

firmware_util_

firmware_util_

Name: firmware_util_

The procedure firmware_util_ contains entry points used to locate firmware programs in a firmware archive.

Entry: firmware_util_\$init

This entry point is used to locate the firmware archive and return a pointer to it which can be used in subsequent calls. The segment "firmware.archive" is searched for, first in the working directory, and then in >ldd>firmware. If found, a pointer to the segment is returned. If not, a standard system status code is returned.

Usage

```
dcl firmware_util_$init entry (ptr, fixed bin(35));  
call firmware_util_$init (fwptr, code);
```

where:

1. fwptr is a pointer to the archive (Output).
2. code is a standard system status code (Output).

Entry: firmware_util_\$find

This entry is used to locate a firmware module in the archive. The module may be identified either by name (useful for itr's or application firmware), or by the ident field of the MPC assembler IDENT pseudo-op card (useful for mdr's).

Usage

```
dcl firmware_util_$find entry (ptr, char(6), char(4), ptr,  
fixed bin(18), fixed bin(35));  
call firmware_util_$find (fwptr, ident, name, modp,  
modlen, code);
```

where:

1. fwptr is a pointer to the firmare archive which was located by firmware_util_\$init (Input).

firmware_util_

firmware_util_

2. ident is the ident field of the module required, or "" if the name (parameter 3) is used (Input).
3. name is the name of the module required, or "" if the ident (parameter 2) is used (Input).
4. modp is a pointer to the start of the firmware module in the archive (Output).
5. modlen is the length of the firmware module, in words (Output).
6. code is a standard system status code. It will be set to error_table_\$noentry if the module cannot be found.

Notes

To locate an itr, or application firmware, the name parameter should be used. For example:

```
call firmware_util_$find (fwptr, "", "btlt", modp,  
modlen, code);
```

To locate an mdr, the ident parameter should be used. For example:

```
call firmware_util_$find (fwptr, "t14aon", "", modp,  
modlen, code);
```

If the firmware module is application firmware, it is the responsibility of the calling procedure to separate it into the control-store overlay and the read-write overlay.

It is also the responsibility of the calling procedure to terminate the firmware archive after all processing is complete.

TITLE: Change meaning of login -po control arg.		STATUS	DATE
AUTHOR: S. Webber		Written	07.29.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.1 -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 08/05/75
		Expires	02/05/76
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input checked="" type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input type="checkbox"/> User Cmmd/Subr.		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	X Commands
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Add a feature to the login control sequence to allow users to specify a prelinked subsystem.

REASONS:

Needed for prelinking and the BASIC and FORTRAN Command and Editing Subsystem.

IMPLICATIONS:

May be an interim solution only, if we decide to change the interface when full prelinking is implemented.

DETAILED PROPOSAL:

Have the answering service scan the argument after the -po (process overseer) control argument and search for the string ",subsystem". If the string is found, take the characters before the string and interpret them as a (relative) pathname to an entry point to be used as the initial procedure to be executed for the process. Specifying ",subsystem" implies ",direct", hence, any prelinked subsystem's initial procedure must perform the initialization done by "init_admin" procedures.

If ",subsystem" is found, the preceding characters are placed in the process overseer fields in the answer_table and the PIT; the bit "dont_call_init_admin" and the new bit "subsystem" are set ON in the PIT and create_info structure used at process creation time.

Once this mechanism is enabled, the hardcore process initialization procedures will interpret the pathname prefix of the process overseer name as a directory in which to find the various template segments left by the pre-linker. These will be used to initialize the process and, later, its rings.

TITLE: Disable access checking for AUTHOR: hphcs_\$set_backup_dump_time Jerry A. Stern		STATUS Written Status Expires	DATE 7/29/75 A 08/05/75 02/05/76
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) _____ -Documented in MFB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input checked="" type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input type="checkbox"/> User Cmmnd/Subr.	DOCUMENTATION CHANGES Document Specify One or More MPM (Vol, Sect.) PLMS (AN #) MOSN (Sect.) MPAM (Sect.) MSAM (Sect.) Info Segs Other (Name) None (Reason) <input checked="" type="checkbox"/>	
Objections/Comments:			

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

Summary: Allow a process having access to hphcs_ (i.e., backup) to set the date/time dumped for an entry without requiring modify permission on the parent directory.

Reasons: The backup process must run with a system_high authorization in order to dump segments and directories of all access classes. This, however, removes effective modify access to all directories of an access class below system_high and thus prevents setting the date/time dumped.

By design, the ability to set the date/time dumped is strictly controlled by the ACL of hphcs_. Furthermore, the date/time dumped field is provided solely for the use of backup. Therefore, removing the requirement for modify access does not in any meaningful way compromise or weaken the ability of a user to control access.

The only alternative to the above scheme is to have the backup process use directory privilege. This, however, is even less desirable since it gives a much broader access privilege to backup that is much more susceptible to accidental abuse.

Implications: None

Detailed Proposal: Change set_\$backup_dump_time to bypass the check for modify access.

TITLE: Multics Sort/Merge		STATUS	DATE
AUTHOR: J. Berson (CEO-Boston/SEC)		Written	02, 20, 75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL		Status	A 08/12/75
-Planned for System MR 3.0		Expires	02/12/76
-Fixes Bug Number(s) _____		DOCUMENTATION CHANGES	
-Documented in MTB (none)		Document	Specify One or More
-User/Operations-visible		MPM (Vol, Sect.)	III, IV
Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no		PLMS (AN #)	(planned)
-Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		MOSN (Sect.)	
-Performance: <input type="checkbox"/> Better <input type="checkbox"/> Same		MPAM (Sect.)	
<input type="checkbox"/> Worse		MSAM (Sect.)	
-Replaces MCR _____		Info Segs	
Objections/Comments:		Other (Name)	
		None (Reason)	

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

SUMMARY:

Proposal for the addition of Sort to the Multics System, Release MR3.0.

The basic function is to Sort a single file to produce an ordered file.

The Sort contains standard input file and output file processing procedures, which support sequential files (segments or multi-segment files) in the Storage System.

Alternatively, the user can specify his own input file and/or output file procedures. Via these procedures, the user can select records for - or delete records from - further processing, either from the input file for processing by the Sort or from the ordered output of the Sort for the output file. Also, the contents of the input file record may be rearranged to form the output file record.

Control Arguments for Command (not included in Section 14.3 of the Subsystem Writers' Guide as of April 30, 1975):

-input_file, -if
-temp_dir, -td
-process_directory, -pd
-replace, -rp

Status Codes and Messages (not included in error_table_ as of January 15, 1974):

data_loss	"Data has been lost."
data_gain	"Data has been gained."

DOCUMENTATION (planned):

sort command, MPM Volume III, Commands

The proposed documents listed above are attached to this MCR.

Sort/Merge Program Logic Manual.

The Sort's standard record comparison procedure supports a single character string key field. Output may be either ascending or descending order. ASCII collating sequence is used. Alternatively, the user can supply his own comparison procedure.

Attached are user specifications, in the form of proposed MPM writeups.

REASONS:

Required by Marketing for a "viable commercial system".
See:

- 1) Multics Sort/Merge Product Functional Specification, #58004067, R. H. Hill, November 6, 1974;
- 2) Multics Sort/Merge Project Summary and Authorization, Project #M64AE;
- 3) Product Calendar Item #MBO233.

IMPLICATIONS:

The following visible names are introduced:

Command:

sort

Entry points

sort_\$release
sort_\$return

Bound Object Segment:

bound_sort_ (not customer-visible)

sort

sort

Name: sort

The sort command provides a generalized file sorting capability that is specialized for each execution by user-supplied parameters. The sort accepts as input a Multics sequential or indexed file in the storage system; ranks its records according to the value contained in a user-specified key field; and produces as output a sequentially organized file in the storage system. The sort command is designed particularly for large files, such as multisegment files. Records may be fixed length or variable length.

Two sets of parameters are associated with a sort execution. One set consists of the arguments to the command, specifying files and selecting processing options. The second set of parameters, called the Sort Description, defines the key field used for ranking records. The Sort Description may be stored as a segment or may be entered directly from the user's terminal when the sort is called.

The user specifies a character-string field as the sorting key. The sort's ranking procedure uses the collating sequence of the Multics standard character set (U.S. ASCII, American National Standards Institute Standard X3.4-1968, hereafter referred to as Multics ASCII). Records are sorted in ascending order unless the user explicitly requests descending order. Records with the same key value maintain their original order; that is, the first of two records on input will be the first on output.

The sort can be further specialized for a specific execution by parameters specifying user-supplied subroutines (called exit procedures), which are then used in place of the sort's standard routines. These parameters are also supplied in the Sort Description. Exit procedures may be supplied to replace any or all of the functions of obtaining input records, ranking records, or producing output.

Usage

```
sort -path- -input_file in_path- -output_file out_path- -control_args-
```

where:

1. path

is the pathname of the segment containing the Sort Description. If this argument is omitted, the -console_input control argument (described below) must be used.

2. -input_file in_path,
-if in_path

is the pathname of the input file. The input file specified must be a sequential or indexed file in the Multics storage system. If the user is supplying his own input file procedure, this argument must be omitted and an input file procedure must be specified in the Exits statement of the Sort Description.

9/75

3-318.3

AG92A

3. `-output_file out_path,` is the pathname of the output file. The output
`-of out_path` file specified must be a sequential file in the Multics storage system. The equals convention may be used; if it is, it is applied against the input file pathname. If the user is supplying his own output file procedure, then this argument must be omitted and the output file procedure must be specified in the Exits statement of the Sort Description. If the `out_path` argument is not specified, the `-replace` argument must be used with the `-output_file` control argument:

`-output_file -replace`

The `-replace (-rp)` argument causes the input file to be replaced by the output file. The input file is overwritten during the merge pass of the sort.

4. `control_args` can be chosen from the following:
- `-console_input, -ci` indicates that the Sort Description is entered through the I/O switch `user_input` (normally the user's terminal).
- `-temp_dir path,`
`-td path` indicates that the user is specifying the pathname of the directory that will contain the work files. The equals convention may be used; if so, it is applied against the input file entryname. If this argument is omitted, work files are contained in the user's process directory. This argument should be used when the user's process directory would not provide sufficient space for the sort's work files.
- `-brief, -bf` suppresses the summary report issued by the sort. If `-brief` is not selected, a report is written on the I/O switch `user_output`.

Notes

The pathname of the segment containing the Sort Description must appear first. Control arguments may appear in any order.

Any pathname can be either relative (to the current working directory) or absolute.

sort

sort

Sort Description Parameters

The Sort Description contains information which specializes the sorting process for a particular execution. This information establishes a key field and/or user-supplied procedures.

A Sort Description is mandatory, since the user must at least specify either a key field or an exit to a user-written compare procedure.

A Sort Description may be supplied either through a segment or through the user's terminal (I/O switch user_input). If the Sort Description is supplied in a segment, its pathname is specified as the first argument of the command.

If the Sort Description is to be supplied through the terminal, the sort prints "Input." through the I/O switch user_output and waits for input. After typing in the Sort Description, the user terminates input by typing a line consisting of a period (.) followed by a carriage return.

Syntax of the Sort Description

A Sort Description is organized into statements. A statement begins with either of the keywords "key" or "exits", optionally delimited by a colon (:). Associated parameters follow, separated by spaces, commas, or in specific cases parentheses. The entire statement is terminated by a semicolon (;).

Key Statement

The key statement specifies the field within an input record whose value is to be used to rank the records on the output file. The statement has the following form:

```
key! datatype(size), position, descending;
```

where:

datatype is a required argument giving the data type of the field. The only data type supported is Multics ASCII character string, represented by the keywords "char" or "character".

2. size

is the field length expressed as a decimal integer.

9/75

3-318.5

AG92A

3. position is the position of the field relative to the beginning of the record. There are two formats:
- w where w is the word offset of the field from the beginning of the record. Words are numbered beginning with zero for the word containing the first character of the record. This format specifies that the key field is aligned on a word boundary.
 - w(b) where w is the word offset as defined above and b is the bit offset of the field within the word. Bits are numbered from 0 to 35. This format implies that the field is not aligned. Although the sorting process functions correctly if it is aligned, speed of execution may be affected.
4. descending, dsc directs the sort to use descending order for this key field. This element may be omitted; the default is ascending order for this key field.

Examples

- | | |
|------------------------|---|
| char(10), 0(18) | Character string, Multics ASCII code, length 10 characters, starting at bit 18 of word 0. |
| char(8), 0, descending | Character string, Multics ASCII code, length eight characters, starting at bit 0 of word 0; use descending ranking. |
| char(4), 0, dsc | Character string, Multics ASCII code, length four characters, starting at bit 0 of word 0; use descending ranking. |

Exits Statement

An exits statement supplies the names of one or more user-written procedures, to be called at the specified exit points during the sorting process.

The exits statement has the following form:

```
exits: exits_description ...;
```


The parameters of the exits statement consist of one or more exit descriptions, each of which is composed of exactly two parameters, which must be written in the specified order. The exit descriptions themselves may be written in any order in the statement.

9/75

3-318.6

AG92A

An exit description has the following form:

exit_name, user_name

where:

1. exit_name is the keyword representing the exit point at which the user-written procedure is to be called. Values may be chosen from the following list:

input_file
output_file
compare

2. user_name is the name of the entry point of the user-written procedure. This parameter has the same syntax and semantics as a command name, as reviewed below.
 - a. The user name can be either a (procedure) segment name (e.g., segment) or a segment name followed by an entry point name (segment\$entry_point). In these cases, the user's current search rules are applied to locate the procedure. (If another segment is already known by the specified reference name, that segment is used.)
 - b. The user name can also be a pathname; that is, it can specify a directory hierarchy location, either relative (to the current working directory) or absolute. In this case, the search rules are not applied and the pathname is used to find the procedure. (If some other segment is already known by the specified reference name, that segment is terminated first.)

Writing an Exit Procedure

The sort process is currently designed to provide three exit points to user procedures. These occur at input file reading, during record comparison, and at output file writing. When the user has indicated that an exit procedure is to be used, the sort calls the user procedure in place of its standard routine at the point specified. The user-supplied procedure must perform the specific function it replaces; that is, an input file procedure must obtain input, and so on. While it has control, however, the user procedure may perform other processing. The user procedure must return to the sort.

The conventions that must be used in writing an exit procedure are given below. Notice that user input and output procedures must call the sort during their operation, as well as return to the sort when they are completed. A user comparison procedure must not call the sort. The sort monitors the sequencing of calls to itself, preventing an improper sequence from being executed.

The entrynames of all exit procedures are defined by the user. Specific names are shown below only for convenience in discussion.

9/75

3-318.7

AG92A

INPUT FILE PROCEDURES

A user-written input file procedure must conform to the specifications given below.

Usage

```
input_file_procedure: proc(code);  
declare code fixed bin(35) parameter;
```

where code is a standard status code that must be returned by the input file procedure. If code is not 0, the sort prints the corresponding message from error_table_ and the sorting process is abandoned.

Structure

For each record that is input to the sort, there must be one call to the sort_release entry point (described below). After the input file procedure has completed, it must return to the sort.

OUTPUT FILE PROCEDURES

A user-written output file procedure must conform to the specifications given below.

Usage

```
output_file_procedure: proc(code);  
declare code fixed bin(35) parameter;
```

where code is a standard status code that must be returned by the output file procedure. If code is not 0, the sort prints the corresponding message from error_table_ and the sorting process is abandoned. (Output)

sort

sort

Structure

For each record to be retrieved there must be one call to the `sort_$return` entry point (described below). If the `sort_$return` entry point is called but there are no more records to be retrieved, then it returns with the status code `error_table_$end_of_info`. This is the normal indication of end of data. If desired, the output file procedure may also terminate prior to receiving the `error_table_$end_of_info` indication from the `sort_$return` entry point. In any case, the output file procedure must return to the sort.

COMPARISON PROCEDURES

A user-specified comparison procedure is called each time the sort is ready to rank two records (that is, to determine which is to be first in the sorted order). The comparison procedure receives as arguments pointers to two buffers, each of which contains a record. The comparison procedure must determine which of the two records is first or that they are equal in rank, and return an appropriate status code to the sort.

Usage

```
compare_procedure: proc(buff_ptr_one, buff_ptr_two)
                    returns (fixed bin(1));

declare (buff_ptr_one, buff_ptr_two) parameter;

declare result      fixed bin(1);
.
.
.
return(result);
```

where:

1. `buff_ptr_one` is a pointer to a buffer containing the first record of the pair to be compared. (Input)
2. `buff_ptr_two` is a pointer to a buffer containing the second record. (Input)
3. `result` is the result of the comparison. (Output) Values are:
0 records rank equal.

- 1 record one ranks first (has lower key values).
- +1 record two ranks first.

Structure

The comparison procedure is invoked as a function. It must return to the sort.

9/75

3-318.9

AG92A

sort

sort

Notes

The sort aligns each record in a buffer area on a double-word boundary. Thus, if the comparison procedure applies a based declaration of the record to the buffer pointers, correct execution is ensured.

If the two records are ranked equal, the sort preserves the original input order of the records.

If the user requires the length (in characters) of either record, it is available in the form:

```
declare rec_length fixed bin(21) aligned;
```

In the word preceding the beginning of the record buffer.

Entry: sort_\$release

The sort_\$release entry point is used each time the user's input file procedure reads a record and releases it to the sort. The caller specifies the location and length of the record. The sort accepts the record and stores it in its own work area. (Portions of the sorting process may also be performed.)

Usage

```
declare sort_$release entry(ptr, fixed bin(21), fixed bin(35));  
call sort_$release (buff_ptr, rec_len, code);
```

where:

1. buff_ptr is a pointer to a byte-aligned buffer containing record. (Input)
2. rec_len is the length of the record in bytes. (Input)
3. code is a standard status code returned by the sort. Possible values are listed below under "Status Codes." (Output)

Notes

The sort aligns each record in a buffer area on a double-word boundary.

9/75

3-318.10

AG92A

sort

sort

Status Codes

The following status codes may be returned by the sort_\$release entry point (all codes are in error_table_):

0	Normal return (no error).
out_of_sequence	The current call is not in the sequence required by the sort; e.g., the sort_\$release entry point has been called after the sort_\$return entry point.
fatal_error	The sort has encountered an error in calling upon some other function of the Multics system, such as the storage system or the I/O system. The sort will have previously printed a specific message related to the condition via the I/O switch error_output.
long_record	This input record is longer than the maximum supported. The record is ignored by the sort, and the caller continue to release records to the sort.
short_record	This input record is shorter than the minimum required to contain the key fields. The record is ignored by the sort, and the caller may continue to release records to the sort.

Entry: sort_\$return

The sort_\$return entry point is used each time the user's output file procedure retrieves a record, in sorted order, from the sort.

The sort returns with the location and length of the record.

If the sort_\$return entry point is called but there are no more records to be retrieved, then the sort_\$return entry point returns to the caller with the status code error_table_\$end_of_info.

Usage

```
declare sort_$return entry(ptr, fixed bin(21), fixed bin(35));
```

```
call sort_$return (buff_ptr, rec_len, code);
```

where:

1. buff_ptr is a pointer to a buffer containing the record. (Output)
2. rec_len is the length of the record in bytes. (Output)
3. code is the standard status code returned by the sort. Possible values are listed below under "Status Codes." (Output)

9/75

3-318.11

AG92A

sort

sort

Notes

The sort aligns each record in a buffer on a double-word boundary. Thus if the caller applies a based declaration of the record to the pointer, correct execution is ensured.

Status Codes

The following status codes may be returned by the sort_\$return entry point (all codes are in error_table):

0	Normal return (not end of information, no error).
end_of_info	There are no more records to be returned from the sort. This is the normal end-of-data condition. No record returned to the caller.
out_of_sequence	The current call is not in the sequence required by the sort; e.g., the sort_\$return entry point has been called before sort_\$release entry point.
fatal_error	The sort has encountered an error in calling upon some other function of the Multics system, such as the storage system or the I/O system. The sort will have previously printed a specific diagnostic message related to the condition via the I/O switch error_output.
data_loss	End of data has been reached, but the number of records previously returned is less than the number of records released to the sort. No record is returned to the caller.
data_gain	The number of records returned (including this record) is now greater than the number of records released to the sort. The current record is returned to the caller, and the caller may continue to request records from the sort.

ort

sort

Examples

The first example given below specifies that the Sort Description is to be entered from the user's terminal.

```
sort -ci -if sort_in -of sort_out  
Input.  
key: char(10), 0;  
.
```

There is one input file, named sort_in; the output file is named sort_out; by default, the work files are contained in the user's process directory; by default, a report is written. The Sort Description in this example provides a Multics ASCII character string keyfield, whose length is 10 characters, starting at word 0 bit 0 of the record. There are no exits specified.

In the following example, the Sort Description is contained in the segment sort_desc.

```
sort sort_desc -td >udd>Alpha>pool -bf
```

Work files are contained in the directory >udd>Alpha>pool; the report is suppressed.

The segment sort_desc contains the following statements:

```
key: char(8) 1;  
exits: input_file user$input,  
output_file user$output;
```

These statements specify that the key is a Multics ASCII character string whose length is eight characters, contained in words 1 and 2 of the record. Two exits are given, an input file procedure exit and an output file procedure exit. The input file procedure is the user's entry point named user\$input; the output file procedure is the user's entry point named user\$output. These exits are mandatory since input file and output file were not provided as command arguments.

In the example below, the Sort Description is again contained in the segment sort_desc.

```
sort sort_desc -lf sort_in -of -replace -fd -wd
```

The input file is named sort_in; the output file is to replace the input file; work files are contained in the user's current working directory; and, by default, a report is written.

9/75

3-318.13

AG92A

Common Termination

Both initial and subsequent attachments allocate storage for the attach description in the cseg chain area and fill it with the string generated by check_attopt. If an IBM DOS file set is being processed, the cseg standard code is updated to 3. Interrupts are then masked, the locb is modified to indicated attachment, interrupts are unmasked, and control returns to the caller.

INTERNAL PROCEDURES

Name: check_attopt

This procedure validates the attach options in the context of either ANSI, IBM SL, or IBM NL file set organization. The sequence of checking is:

1. Physical medium characteristics: number of tracks and density. These checks are common to all contexts.
2. Context-specific checks: mutual exclusivity of ANSI and -dos options, etc. Each type has some particular combination of options that must be validated.
3. Output mode checks: mutual exclusivity of -extend and -expires options, etc. Each output mode requires the presence or absence of other options.
4. Record format checks. Each logical record format places particular constraints upon the file attribute options.

Checks are performed in an order that minimizes superfluous processing.

TITLE: Install dump_mpc command		STATUS	DATE
AUTHOR: L. Johnson		Written	07.16.75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR -Fixes Bug Number(s) -Documented in MTB -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR	Category (Check One)		Status
	<input type="checkbox"/>	Lib. Maint. Tools	Per/29/75 A 08/12/75
	<input checked="" type="checkbox"/>	Sys. Anal. Tools	Expires
	<input type="checkbox"/>	Sys. Prog. Tools	02/10/76
	<input type="checkbox"/>	355	DOCUMENTATION CHANGES
<input type="checkbox"/>	BOS	Document	Specify One or More
<input type="checkbox"/>	Salvager	MPM (Vol, Sect.)	
<input type="checkbox"/>	Ring Zero	PLMS (AN #)	
<input type="checkbox"/>	Ring One	MOSN (Sect.)	
<input type="checkbox"/>	SysDaemon/Admin.	MPAM (Sect.)	
<input type="checkbox"/>	Runtime	MSAM (Sect.)	
<input type="checkbox"/>	User Cmnd/Subr.	Info Segs	
Objections/Comments: Documentation should mention that the command requires access to the gate rcp_priv_ and that the output is similar to the BOS MPCD command.		Other (Name)	Hardware Diagnostic Aids
		None (Reason)	

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

SUMMARY:

Install on-line version of dump_mpc command which performs a dump of mpc read/write memory and an edit of the mpc trace table. This command performs the same function as the BOS MPCD command.

REASONS:

This is a useful tool for analyzing I/O problems. It should be available without having to crash the system to use the BOS version.

DETAILED PROPOSAL:

See attached command description.

Name: dump_mpc

The dump_mpc command will perform a dump of the read/write memory of a microprogrammed peripheral controller (MPC) and edit the dump, the trace table, and device statistics.

Usage

dump_mpc device -path-

where:

1. device is the name of the MPC to dump, or in the case of a unit record MPC, the name of a device connected to it.
2. path is an optional pathname of a segment where the dump is to be stored. The default is mpc-type.list where mpc-type is "tape", "disk", or "urc".

In order to dump a tape MPC, the device should be specified as "tape", or "tape_00". Similarly, for disk, the device should be "disk", or "disk_00". To dump a disk MPC, however, the system must be configured with an I/O disk connected to the MPC. To dump a unit record MPC, any device connected to the MPC may be specified.

Before using this command, it may be necessary to place switch 4 on the MPC in the up position. This inhibits any further tracing of I/O in the MPC.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR <u>1304</u>
TITLE: Change dump_segment not to dump ring 0 segments. AUTHOR: Steve Webber		STATUS DATE Written 07/31/75 Status 08/02/75 Expires 01/31/76
Planned for System: not applicable Fixes Bug Number(s): not applicable Documented in MTB: not applicable Incompatible Change: yes User/Operations-visible Interface Change: yes Coded in: (<input checked="" type="checkbox"/>)PL/I (<input type="checkbox"/>)ALM (<input type="checkbox"/>)other-see below Performance: (<input type="checkbox"/>)better (<input checked="" type="checkbox"/>)same (<input type="checkbox"/>)worse Replaces MCR: 1304		CATEGORY (check one) (<input type="checkbox"/>)Lib. Maint. Tools (<input type="checkbox"/>)Sys. Anal. Tools (<input type="checkbox"/>)Sys. Prog. Tools (<input type="checkbox"/>)355 (<input type="checkbox"/>)BOS (<input type="checkbox"/>)Salvager (<input type="checkbox"/>)Ring Zero (<input type="checkbox"/>)Ring One (<input type="checkbox"/>)SysDaemon/Admin (<input type="checkbox"/>)Runtime (<input checked="" type="checkbox"/>)User Command/Subr
DOCUMENTATION CHANGES (specify one or more) MPM (vol,sect) Commands MPAM (sect) MOSN (sect) MSAM (sect) PLMs (AN#) AN51 Info Segs Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

1. Add entries ring_zero_dump and rzd to dump_segment which will allow dumping ring-zero segments - otherwise the same as dump_segment.
2. Delete the program ring_zero_dump from >tools.
3. Add new control arguments:
 - address, -no_address
 - offset, -no_offset
 - header, -no_header
 - block

REASONS:

1. Users should not, in general, be able to dump ring-zero segments accidentally even though they may have access to. Dumping ring-zero segments should be done explicitly.
3. Convenient features. See attached documentation which is inserted into the standard MPM documentation for dump_segment.

IMPLICATIONS:

Better user interface (back to what it was before latest dump_segment was installed with respect to which segments are dumped).

DETAILED PROPOSAL:

The following is to be inserted into the MPM documentation for `dump_segment` at the appropriate place.

- `-header, -he` is used to have a header line containing the pathname (or segment number) of the segment being dumped as well as the date/time printed. The default is not to print a header unless neither "first" nor "count" are specified, i.e., unless the entire segment is being dumped.
- `-no_header, -nhe` is used to suppress printing of the header line even though the entire segment is being dumped.
- `-address, -ad` causes the address (relative to the base of the segment) to be printed with the data. This is the default.
- `-no_address, -nad` causes the address not to be printed.
- `-offset n, -ofs n` causes the offset (relative to n words before the start of data being dumped) to be printed along with the data. If n is not given, 0 is assumed.
- `-no_offset, -nofs` causes the offset not to be printed. This is the default.
- `-block m` causes words to be dumped in blocks of m words separated by a blank line. If the `-offset` control argument is specified, its parameter, n, is ignored, if given, and 0 is used. The offset, if printed, is reset to zero at the beginning of each block printed.

The documentation for `ring_zero_dump` (in the Tools PLM) will be identical to the MPM documentation for `dump_segment` except for the following addition (at the appropriate place):

This command can be used to dump ring-zero segments as well as user-ring segments. If a segment number is used to specify the segment to be dumped, that segment will be dumped (if the user has access). If a name is specified, a search of the hardcore segment names (as defined by the SLT) is made, and, if the search is successful, the specified ring-zero segment is dumped. If the name is not found in the SLT, it is treated as a (relative) pathname of the segment to dump (in the same manner that `dump_segment` does it).

TITLE: Supply "write_status" order for MCS		STATUS	DATE	
AUTHOR: R. Coren		Written	07.30.75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR <u>3.0</u> -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)	Status	A 08/12/75	
		Lib. Maint. Tools	Expires	02/12/76
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355	MPM (Vol, Sect.)	AG93
	BOS	PLMS (AN #)		
	Salvager	MOSN (Sect.)		
	X Ring Zero	MPAM (Sect.)		
	Ring One	MSAM (Sect.)		
	SysDaemon/Admin.			
	Runtime			
	User Cmnd/Subr.			
Objections/Comments:		Info Segs		
		Other (Name)		
		None (Reason)		

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

SUMMARY:

Provide a "write_status" order for MCS which informs the caller whether there is output still pending in the ring zero buffers, and if so, wakes up the process when the output completes (more or less parallel to "read_status").

REASONS:

The new g115_ DIM needs to send complete output messages one at a time, and therefore must not send a message until the previous one has been shipped to the 355.

IMPLICATIONS: None.

Note: Since the immediate need for this feature is an internal interface, it is proposed not to update the MPM documentation on tty_ for MR3.0, but to do so at our leisure for a future release.

(Add to list of control operations for tty_ in MPM subr.)

`write_status` is used to ascertain whether any output for the terminal is pending in ring zero buffers. The `info_ptr` should point to the following structure:

```
dcl 1 wr_stat aligned,  
2 ev_chan fixed bin(71),  
2 output_pending bit(1);
```

1. `ev_chan` is the name of an event channel over which the process will receive a wakeup when pending output has been sent; it is not used unless `output_pending` is "1"b. (Output)
2. `output_pending` is set to "1"b if there is output still in the ring zero buffers; otherwise it is set to "0" b. (Output)

Multics Change Request

TITLE: Supply "set_line_type" order for MCS		STATUS	DATE
AUTHOR: R. Coren		Written	07.30.75
-Coded in: <input type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 08/12/75
		Expires	02/12/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Supply a control operation, "set_line_type", to the tty_DIM. This will allow the line type (used by the 355 to determine line control) to be modified by Multics.

REASONS:

MCS must be informed by the Answering Service of the line types of certain lines that cannot be otherwise determined, e.g., dialup high-speed synchronous lines and TN1200 on 202C6 modem.

IMPLICATIONS: None.

DETAILED PROPOSAL:

Modify tty_index to recognize the new order; supply a 355 mailbox opcode to pass the order to the 355, and modify 355 modules to recognize it and act accordingly.

Note: Since the immediate need for this feature is as an internal interface, it is proposed not to update the MPM documentation on tty_ for MR3.0, but to do so at our leisure for a future release.

(Add to list of tty_ control operation in MPM subr.)

`set_line_type` causes the line type associated with the terminal to be set to the value supplied. The `info_ptr` should point to a fixed bin variable containing the new line type. This operation is not permitted for a line that is dialed up.

Multics Change Request

TITLE: Fix pxss processid checking bug		STATUS	DATE	
AUTHOR: R. Mullen		Written	08.04.75	
-Coded in: <input type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR <u>3.0</u> -Fixes Bug Number(s) <u>unreported</u> -Documented in MTB -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status <u>A 08/12/75</u>	
	<input type="checkbox"/>	Lib. Maint. Tools	Expires <u>02/12/76</u>	
	<input type="checkbox"/>	Sys. Anal. Tools	DOCUMENTATION CHANGES	
	<input type="checkbox"/>	Sys. Prog. Tools	Document	Specify One or More
	<input type="checkbox"/>	355		
	<input type="checkbox"/>	BOS	MPM (Vol, Sect.)	
	<input checked="" type="checkbox"/>	Salvager	PLMS (AN #)	
	<input checked="" type="checkbox"/>	Ring Zero	MOSN (Sect.)	
	<input type="checkbox"/>	Ring One	MPAM (Sect.)	
	<input type="checkbox"/>	SysDaemon/Admin.	MSAM (Sect.)	
<input type="checkbox"/>	Runtime			
<input type="checkbox"/>	User Cmmd/Subr.			
Objections/Comments:		Info Segs		
		Other (Name)		
		None (Reason)		

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

SUMMARY:

Fix processid checking code in pxss to not take overflow fault.

REASONS: Overflow fault is fatal.

IMPLICATIONS: None

DETAILED PROPOSAL:

Reject negative process_id's immediately.

Ver. 3	MULTICS CHANGE REQUEST	MCR 1333
741022		
TITLE: Fix bug in reduction_compiler_		STATUS DATE
		Written 08/04/75
AUTHOR: Gary C. Dixon		Status A08/12/75
		Expires 02/04/76
Planned for System: not applicable		
Fixes Bug Number(s): unreported bug		CATEGORY (check one)
Documented in MTB: not applicable		() Lib. Maint. Tools
Incompatible Changes: no		() Sys. Anal. Tools
User/Operations-visible Interface Change: no		(<input checked="" type="checkbox"/>) Sys. Prog. Tools
Coded in: (<input checked="" type="checkbox"/>) PL/I () ALM () other-see below		() 355
Performance: () better (<input checked="" type="checkbox"/>) same () worse		() BOS
		() Salvager
DOCUMENTATION CHANGES (specify one or more)		() Ring Zero
MPM (vol,sect) MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		() SysDaemon/Admin
PLMs (AN#)		() Runtime
Info Segs		() User Command/Subr
None (reason) No interface changes made, only bug fix.		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Fix bug which causes the reduction_compiler_ subroutine of the reduction_compiler command to optimize storage for tokens incorrectly.

REASONS:

This bug can cause bad code in the translators generated by the reduction_compiler.

IMPLICATIONS:

A bug will be fixed.

Multics Change Request

TITLE: Fix bug in io_call command		STATUS	DATE
AUTHOR: L. Johnson		Written	08.04.75
<input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 08/12/75
		Expires	02/12/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input checked="" type="checkbox"/> User Cmnd/Subr.		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason) will work as docu- mented	

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

SUMMARY:

If a get_line or get_chars operation requests more data than remains to be input over an i/o switch, the iox_ call will return the remaining data along with an error code. In this case, the io_call command prints the error, but ignores the data.

DETAILED PROPOSAL:

Change the command to do what normally would have been done to the data (print it or store it in a segment) before printing the error message.

Ver. 3	MULTICS CHANGE REQUEST	MCR 1335
741022		
TITLE: Modify Answering Service for MCS Phase 2	STATUS	DATE
AUTHOR: Paul Green	Written	07/31/75
	Status	A 08/12/75
	Expires	01/31/76
Planned for System: MR 3.0	CATEGORY (check one)	
Fixes Bug Number(s): not applicable	() Lib. Maint. Tools	
Documented in MTB: 171	() Sys. Anal. Tools	
Incompatible Change: no	() Sys. Prog. Tools	
User/Operations-visible Interface Change: yes	() 355	
Coded in: () PL/I () ALM () other-see below	() 80S	
Performance: () better () same () worse	() Salvager	
DOCUMENTATION CHANGES (specify one or more)		
MPM (vol,sect) MPM Com. MPAM (sect)	() Ring Zero	
MOSN (sect) 6.2.2 MSAM (sect) ✓CMF	() Ring One	
PLMs (AN#)	() SysDaemon/Admin	
Info Segs login.info	() Runtime	
Other	() User Command/Subr	
OBJECTIONS/COMMENTS:		
SRB description of CMF Data set information in MOH.		

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

SUMMARY:

Implement the Channel Master File as described in MTB-171, and as amended in this MCR. Specifically, provide

1. An ASCII Channel Master File, and a program cv_cmf to compile it into a Channel Definition Table.
2. Code in answering service initialization to initialize the answer table from the CDT instead of from the lines file. (An exec_com will be provided to convert the lines file into a CMF for installation purposes.)
3. Answering service support for dial-up G115's.
4. Answering service command "MAP" for all terminals, not just network.
5. Answering service support for correspondence-code 2741's.
6. Answering service support for TN1200's on 202c6 datasets.

The extended dial facility and request dispatcher described in MTB-171 have already been implemented.

REASONS:

Several new capabilities are being added to the teletype DIM and MCS software for MR 3.0. These changes require support in the answering service.

PROPOSAL:

The initial version of the Channel Master File (CMF) will allow an administrator to specify the following attributes per channel:

- channel name
- access class
- audit control flag
- expected answerback id
- initial terminal and line types
- 2741 code selection
- process group id of owner

Two new login control arguments (-terminal_type and -modes) will also be added. See the draft documentation attached to this MCR.

A new pre-login command named MAP will be documented (it has always been there, for the network). It will now work for non-network terminals, as well. See the draft documentation attached to this MCR.

Two more new pre-login commands will allow use of Correspondence or EBCDIC 2741's. These commands will be named after the type element normally used for the code set. Actually there will be four commands:

963	no operation
015	no operation
914	switch to Correspondence code
063	switch to EBCDIC code

Support for dial-up G115's and TN1200's on 202c6 modems will be accomplished by setting the initial line discipline in the CDT entry for the channel.

ADDITIONAL CONTROL ARGUMENTS FOR THE LOGIN COMMAND:

-terminal_type devname

-tftp devname

where devname specifies the terminal that the user is using. This control argument overrides the default terminal type. devname may be chosen from the following list:

1050	an IBM Model 1050
2741	an IBM Model 2741, EBCDIC codes
CORR2741 or corr2741	an IBM Model 2741, Correspondence codes
TTY37 or tty37	a device similar to a Teletype Model 37
TTY33 or tty33	a device similar to a Teletype Model 33 or 35
TTY38 or tty38	a device similar to a Teletype Model 38
TN300 or tn300	a device similar to a TerMINet Model 300 or 1200
ARDS or ards	a device similar to an ARDS
ASCII or ascii	a device similar to a CDI 1030 or TI 725

-modes modestring

This control argument causes the terminal mode to be set to "modestring" after the login process has completed. Consult the MPH description of the tty_ I/O module for information on terminal modes.

THE MAP PRE-LOGIN COMMAND:

Name: MAP, map

This command can be given before the login, enter, or enterp commands. It changes the translation tables used by the terminal control software so that all input will be treated as if the device was a Teletype Model 33 or 35. That is, all upper case alphabetic characters will be translated to lower case, and special escape conventions will be interpreted for ASCII graphics which are not on the TTY33 keyboard.

This command must be used for 150 and 300 baud terminals if the terminal can transmit only upper case; the MAP command is invalid on a 2741 or 1050-type terminal.

For terminals which have no answerback, the system makes a default assumption about the terminal type based on the baud rate of the channel.

Speed	Default
110	TTY33 (MAP mode)
133	2741
150	TTY37
300	ASCII

MULTICS HANDLING OF EBCDIC AND CORRESPONDENCE CODE 2741's:

Multics is capable of handling both EBCDIC and Correspondence 2741's. All 133-baud channels will support login from either type of terminal. The system administrator specifies which code set is to be the default for each channel. The user who dials up the "wrong" channel will receive a greeting message which is gibberish. He should then issue the "015" command if he has a Correspondence terminal, or the "963" command if he has an EBCDIC terminal. For 133-baud terminals, the "Type 'help' for instructions" message will be augmented by a message which is one of

tidu #63 cqn U1XV0XK Type 015 for Correspondence code.
Type 963 for EBCDIC. U1a; z17 q1s F1ss;na1rpi;rf; fipi-

TITLE: Move G115 protocol support to 355		STATUS	DATE
AUTHOR: M. Grady		Written	08.05.75
<input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input checked="" type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	H 08/12/75
		Expires	02/12/76
		DOCUMENTATION CHANGES	
		Lib. Maint. Tools	
		Sys. Anal. Tools	
		Sys. Prog. Tools	
		X 355	Document Specify One or More
		BOS	
		Salvager	MPM (Vol, Sect.)
		Ring Zero	PLMS (AN #) 85
		Ring One	MOSN (Sect.)
		X SysDaemon/Admin.	MPAM (Sect.)
		Runtime	MSAM (Sect.)
		User Cmnd/Subr.	
Objections/Comments:			Info Segs
			Other (Name)
			None (Reason)

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

SUMMARY:

Move G115 protocol support to 355. Add support for dialup sync modems.

REASONS:

Move of G115 protocol support will:

1. make more effective use of the 355 processor and communications abilities;
2. reduce the load on Multics when using remote printers.

Support for dialup modems has been requested many times by various customers/MIT.

DETAILED PROPOSAL:

Coded in 355 map.

Multics Change Request

TITLE: Support Terminet 1200 using Bell 202C5/6 modems
AUTHOR: M. Grady

STATUS	DATE
Written	08.05.75
Status	A 08/12/75
Expires	08/19/76

-Coded in: PL/I ALM other-
explain in DETAILED PROPOSAL
-Planned for System MR 3.0
-Fixes Bug Number(s) unreported
-Documented in MTB _____
-User/Operations-visible
Interface change? yes no
-Incompatible change? yes no
-Performance: Better Same
 Worse
-Replaces MCR _____

Category (Check One)	
<input type="checkbox"/>	Lib. Maint. Tools
<input type="checkbox"/>	Sys. Anal. Tools
<input type="checkbox"/>	Sys. Prog. Tools
<input checked="" type="checkbox"/>	355
<input type="checkbox"/>	BOS
<input type="checkbox"/>	Salvager
<input type="checkbox"/>	Ring Zero
<input type="checkbox"/>	Ring One
<input type="checkbox"/>	SysDaemon/Admin.
<input type="checkbox"/>	Runtime
<input type="checkbox"/>	User Cmnd/Subr.

DOCUMENTATION CHANGES	
Document	Specify One or More
MPM (Vol, Sect.)	
PLMS (AN #)	
MOSN (Sect.)	
MPAM (Sect.)	
MSAM (Sect.)	
Info Segs	
Other (Name)	description of how terminal works
None (Reason)	X

Objections/Comments:
Need paragraph in SRB

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

SUMMARY:

Provide support for Terminet 1200 using coded line control with the Bell 202C5/6 modems.

REASONS:

A number of customers have requested such support.

IMPLICATIONS:

For the first time a site will be required to specify the intended use of a particular modem, rather than have the system guess or determine by default the intended use. (Both the ARDS and the Terminet may use a 202C6 modem, but the system must know before such use which is intended since it would be impossible to tell by guess or experiment.)

DETAILED PROPOSAL:

Support will be provided for the TN1200 in this mode of operation. The terminal will be clumsy to the average Multics user since it will not permit read-ahead or output during user input. The terminal is placed into input mode when the user's process does a read, and will terminate when an etx is sent by the user. The terminal will then remain in output mode until the user's process does another read.