MULTICS TECHNICAL BULLETIN MTB-005

DATE: October 23, 1973

TO: Distribution

FROM: M. S. Hodges

SUBJECT: Multics Change Requests

Attached are copies of all Multics Change Requests which were approved from October 1 to October 15, 1973.

Version 1
730822

MULTICS CHANGE REQUEST	Page 1 of 1 Pages

OBJECTIONS RECORDED:

STATUS	DATE
Written	9/14/73
Initial Approval #	10/9/73
Initial Rejection	
Postponed	9/18/73
Withdrawn	
Expiration Date	

MCR 009

AUTHOR: N. I. Morris

SOURCE: (if external) e.g., "User", "Marketing"

Marketing

	CLASSIFICATION		JUSTIFICATION
X	Incompatible Change	X	Marketing Requirement
X	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
			Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Comment:

Refer to MCR-007

Subject Justification Summary (Detailed Proposal)

SUBJECT: Modify BOS to handle tapes on the MTS-500

The BOS bootstrap loader (which was on 1 card and is now on 4) must be modified to select the correct drive density and allow drive selection (since MTH-500's cannot be redialed to different drive numbers or different densities).

All BOS programs which read or write tape are affected. Special primitives are being supplied which will select the correct density. All tape DCW lists have been modified to allow compatibility between MTS-400 and MTS-500.

In addition, all programs which read or write Multics standard tapes will be made to handle 1040 word records.

Most of the work has already been completed.

ersion 2
30919

MULTICS CHANGE REQUEST	MCR 021 Page 1 of 1 Pages

TITLE: Modify linker and add new stack frame flags

AUTHOR: M. Weaver

SOURCE: (if external) e.g., "User", "Marketing"

Local

STATUS	DATE
Written	9/17/73
Approved msh	10/4/73
Rejected	
Postponed	9/25/73
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension	X	Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement	X	Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

Postponed for clarification.

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

REASONS:

1) When the linker is invoked because of a fault_tag_2 on a non-link, it either crashes the system or tries to proceed and encounters garbage.

2) The linker currently handles trap-before-link and trap at first reference by the condition mechanism, signalling them as linkage_errors. They are not in fact conditions and should use a separate mechanism.

3) New stack frame flags need to be added for the changes to the condition mechanism and for (2). The signaller flag should be moved as it is in the bit offset field of the forward pointer.

4) signal_ soon needs to recognize the "any_other" condition. This is not related to others but signal_ needs to be changed for them anyway.

SUMMARY: The linker is changed to detect most cases of illegal fault tag 2's. For traps before link and traps at first reference the linker calls a special procedure rather than using the condition mechanism. This procedure creates a stack frame in the proper outer ring stack and transfers to a procedure in the outer ring which invokes the real trap procedure (and then transfers back to where the linker was invoked). This change enables has_$make_ptr to process traps at first reference, which will allow bound segments with first reference traps to be called for the first time from command level (when the binder produces proper first reference traps).

The procedure used by the unwinder is made externally available so the linker can use it and it and the interface is made more general.

signal_ (which is also the unwinder) is changed because of the change to its utility procedure, because of the changes in the stack frame flags (3 of which affect it specifically) and to recognize the "any_other" condition. Both new and old crawlout flags are set so that existing system debugging tools will work correctly.

signaller is changed to set the new signaller flag as well as the old, for compatibili

All stack frame flags will be in the first word of the back pointer.

IMPLICATIONS: All of these changes have been implemented and are ready for installation. The only immediate notification to users should be about the any_other condition.and the changes in the stack frame flags. For the rest, there need to be the appropriate MPM (error_table_), SPS and SWG updates. These changes should be transparent.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 033
Page 1 of 1 Pages

TITLE: New command/active function "user" and new command/active function "system"

AUTHOR: T. H. VanVleck

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/20/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement	X	Simplification
	Reliability Improvement	X	Generalization
			Bug Fix

Replaced by proposal MCR ______

Implemented in System ______

Objections/Comments:

Document as active functions only (not as commands).

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

REASONS: Active functions cause stack to be capped and two error messages to be typed if issued as commands; this is foolish. Also, as exec_com programming increases, we find that we have a subroutine, active function, and command for every data item the user wishes to retrieve. This proposal is a modest step toward eliminating a little of the unnecessary command system complexity.

SUMMARY: If the "user" active function is called, by the command processor, it works as it does now, and returns the selected user parameter to the command processor. If "user" is called as a command, it detects this by getting "error_table_$ not_active_fnc_" and instead of causing a big fuss, simply types its value.

Two new parameters have also been added to the "user" command: "outer_module" and "process_id", primarily for use by exec_coms.

The "system" command/active function is new. It is like "user" but returns per-system parameters, such as "sysid", "n_users", "last_down_time", etc.

IMPLICATIONS: This change is upward compatible. It does establish a precedent, that a program may be both a command and active function; I don't see that it requires that all active functions be commands. How we list these hybrid programs in the MPM may be a question.

ersion 2
30919

MULTICS CHANGE REQUEST	MCR 034 Page 1 of 1 Pages

TITLE: **Install new billing program**

AUTHOR: T. H. VanVleck

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/19/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension	•	Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: This new billing program, already in use locally, produces an additional copy of the monthly usage report with separator pages suitable for interdepartment mailing. It also produces an additional page per project showing charges for ARDS usage, tape usage (when charged), etc.

SUMMARY: Modifications to "write_user_usage_report" and biller.ec, and new program "mailing_page_".

IMPLICATIONS: Slightly better bill. More storage needed for monthly billing run.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 035
Page 1 of 1 Pages

TITLE: New commands for directory auditing

AUTHOR: T. H. VanVleck

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/20/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement	X	Generalization
			Bug Fix

Replaced by proposal MCR ________

Implemented in System ________

Objections/Comments:

In Tools library only and document in SPS only; installation/maintenance aids.

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

REASONS: These commands allow a user to snapshot a directory and to determine the difference between two snapshots. These commands may be used for crash recovery, damage assessment, and security auditing.

SUMMARY: There are four commands:

save_dir_info path -seg-
This command lists all of path into the segment seg.dir_info. Everything obtainable from the file system is saved.

comp_dir_info seg1 seg2 -ca-
This command compares two dir_info segments. Differences are reported. ca may be "-bf" or "-lg"

list_dir_info seg -ca-
This command lists a dir_info segment.

rebuild_dir seg -ca-
This command takes a dir_info segment and compares it to the current version of the directory. If a directory or link is missing, it is re-created. If a segment is missing, the info on the segment is typed unless "-bf" was specified.

ersion 2
30919

MULTICS CHANGE REQUEST

MCR 036
Page 1 of 1 Pages

TITLE: Lay trap for reused address bug

AUTHOR: S. Webber

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/21/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix
		X	Find Bug

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: To help find the reused address bug, another trap was placed in the system.

SUMMARY: A check is made when storing into a critical field of a PDME that one of the index registers has not been clobbered.

NOTE: Installed 9/20/93 as emergency installation.

Version 2
730919

MULTICS CHANGE REQUEST	MCR 037 Page 1 of 1 Pages

TITLE: Simplification of command loop interface to users

AUTHOR: M. Weaver

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/20/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement	X	Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: The current action of the system after a quit or condition is often misleading or confusing. The user may not realize that he is not at his original command level and if he types anything other than hold, most of his stack will disappear upon return. Some users, after typing more quits than starts, do not realize why their computations do not get completed.

Another inconsistency is that a condition "wall" is established after a quit or condition, cutting off all the on-units before it is in the stack, while everything else in the process, such as internal static, is unchanged. It has seemed to cause more problems than it solved.

release and start do not behave as other commands, but take effect only after the command lines in which they are located have been completely processed.

SUMMARY: The condition wall will no longer be established.

The automatic release will be discontinued, so every release will be explicitly requested. In order to inform the user that he is not at his original command level and that his stack is building up, the ready message, when not at the first command level, will print the level number of the current command level and the stack frame number of the caller of the ready procedure. This should also make mismatched quit-starts quite obvious.

release and start will take effect immediately, with any part of the command line following them being thrown away.

This is primarily a user command level interface change, so while users must be informed ahead of time, they shouldn't have to change many programs. The speed of the basic command loop with ready messages will be a little slower, and each new invocation of the listener will take a little longer.

DETAILED PROPOSAL: All necessary changes to system routines have been implemented. The installation-maintained routine general_ready is widely used and so should be updated also. However, since it is sometimes used in subsystems that have their own listener, it should not automatically print the level number. A new control argument, -level, will be added to specify printing of the level and frame numbers. An external entry in listen_ will be provided for obtaining the level and frame numbers. This information is in listen_'s automatic storage and is the only such information needed externally. This listen_entry should be provided before the rest of the changes, returning in all cases level=1, frame-0 so that a compatible version of general_ready can be installed ahead of time.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 038
Page 1 of 1 Pages

TITLE: **Modification of hardcore MSL routines to add source types "mx" (.mexp) and "mt" (.mt).**

AUTHOR: **D. Jordan**

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/19/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ______

Implemented in System ______

Objections/Comments:

Install in tools library.

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

REASONS: **Currently installed routines do not handle these relatively recent source types, thus resulting in a loss of information from the hardcore MSL.**

SUMMARY: **This change requires minor editing of the tools procedure msl_util1.pl1, (bound_msl_). This change does not include conversion of any msl procedures to Version 2 PL/1 as our resources are limited.**

IMPLICATIONS: **A special msl_transmog will be done on the hardcore msl. This will be performed by the implementor at the time this change is installed.**

Version 2
730919

MULTICS CHANGE REQUEST

MCR 039
Page I of I Pages

TITLE: Modify accounting package to check that total virtual CPU charged equals total virtual CPU used.

AUTHOR: T. H. VanVleck

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/19/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ________

Implemented in System ________

Objections/Comments:

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

REASONS: This check insures that system usage records are consistent. The check is commented out in act_ctl_ due to bugs in total CPU measurement which have now been fixed. Error message is same as before; already documented.

SUMMARY: minor changes to as_meter_ and un-commenting of code in act_ctl_.

IMPLICATIONS: None, unless there is a concealed bug in total CPU accounting, in which case this will expose the bug by logging error messages.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 042
Page 1 of 1 Pages

TITLE: Fix bug causing lost status from 355

AUTHOR: R. B. Snyder

SOURCE: (if external) e.g., "User", "Marketing"

local

STATUS	DATE
Written	9/21/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ______

Implemented in System ______

Objections/Comments:

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

REASONS: A bug was discovered wherein status in the 355 which was to be sent to the 6180 was being delayed or lost.

This bug was fixed, submitted and is installed. It was done because it was an "emergency fix", i.e., one which was causing crashes.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 043
Page 1 of 1 Pages

TITLE: **Fix bugs in system_control_**

AUTHOR: T. H. VanVleck

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/20/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
X	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: Some bugs in system_control_ seem to cause the initializer to lock up. Errors during answering service startup cause the system to stop initialization but the system continues on to the next step in some cases. This change fixes these bugs and adds the new function of allowing the operator to list the routing tables.

SUMMARY: Modifications to system_control_. as_init_ will be changed, after system_control_ is installed, to return an error code.

IMPLICATIONS: None.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 044
Page 1 of 1 Pages

TITLE: Remove >tools from default search path

AUTHOR: T. H. VanVleck

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/20/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
X	Incompatible Change		Marketing Requirement
	Extension	X	Conformance to Standard
X	Restriction	X	Increased Consistency
X	Performance Improvement	X	Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: Users sometimes invoke programs in the TOOLS library accidentally as the result of a typing mistake. They also pay for searching this library when looking for missing segments.

This change has been planned for a long time. Everything finally seems ready.

SUMMARY: active_hardcore_data.alm will be modified to remove >tools. A development run will be made to insure that the system will boot and system processes will run.

IMPLICATIONS: Users should be warned, via pending_changes.info.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 045
Page 1 of 1 Pages

TITLE: Avoid Reused Address Problems

AUTHOR: Steve Webber

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/21/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ________

Implemented in System 20.11j

Objections/Comments:

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

REASONS: Fix code in page control to repair damage done by reused address bug.

SUMMARY: The reused address bug has not been found but page control can find the results of the bug before any damage is done. The problem can then be fixed so the system continues running. No message is printed when the damage is repaired.

IMPLICATIONS: This takes pressure off of finding the reused address bug, but we should still search dilligently.

Version 2
730919

MULTICS CHANGE REQUEST	MCR 046 Page 1 of 1 Pages

TITLE: **Merge privileged and unprivileged code in ALM kernel of page control.**

AUTHOR: **Steve Webber**

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/21/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
X	Performance Improvement	X	Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: **The combining of page_fault and the wired_fim handler for page faults will simplify and speed up the page fault mechanism. The new combined program will be privileged, but with the new hardware this does not permit page fault to do any harm such as storing into read-only segments. All it does is allow the execution of privileged instructions.**

SUMMARY: **page_fault and the parts of wired fim and master_pxss_page used by page fault should be merged.**

IMPLICATIONS: **either bound_page_control will have to be made privileged or a new bound segment should be created which will be privileged.**

Version 2
730919

MULTICS CHANGE REQUEST

MCR 047
Page 1 of 1 Pages

TITLE: **Merge privileged and unprivileged code in the traffic controller**

AUTHOR: **Steve Webber**

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/21/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
X	Performance Improvement	X	Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: **See arguments for MCR-046**

SUMMARY: pxss and master_pxss_page should be merged together.

IMPLICATIONS: Since pxss and master_pxss_page are not bound, no new bound segments need be created.

ersion 2
30919

MULTICS CHANGE REQUEST

MCR 048
Page 1 of 1 Pages

TITLE: Add new mode to TTY DM for APL

AUTHOR: Paul A. Green

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9-25-7
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
	Extension	•	Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS:

Add "hndlquit" mode (default on) which controls automatic resetread and automatic typing of new line upon the QUIT key being pressed. Turning this mode off will allow us to implement an APL/360 compatible editor. The Multics MACLISP interpreter also needs this mode to implement an ITS - compatible MACLISP.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 050
Page 1 of 1 Pages

TITLE: fix bug in full-command-processor-

AUTHOR: Paul A. Green

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9-25-73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
X	Incompatible Change		Marketing Requirement
	Extension	X	Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement	X	Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS:

Presently, the command lines:

```
XXX ( )
XXX ( [active_function])
```

do not behave identically if the active function returns a null character string. This change fixes the bug so that they behave identically for this case.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 051
Page 1 of 1 Pages

TITLE: Change prt-300-conv to fix bug

AUTHOR: Paul A. Green

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9-25-'
Approved #	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
X	Incompatible Change		Marketing Requirement
	Extension	•	Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS:

This program handles code conversion for the PRT 300 line printer. Currently, it very carefully does not print "non-printing" ASCII control characters, but very carelessly maps non-ASCII characters > 177(8) into ASCII by making off the high order bits. This change causes it to handle characters > 177(8) the same as non-printing control characters.

# changed to 6180 opcodes as well.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 053
Page 1 of 1 Pages

TITLE: Extended Star/Equal Convention
(Online Library Portion)

AUTHOR: Gary C. Dixon

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9-25-73
Approved #	10/9/73
Rejected	
Postponed	10/4/73
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement	X	Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS:

1) Complete the implementation of the extend Star/Equal Convention, as described and approved in MSB-103. (See attachment)

Summary:

1) Install new versions of check_star_, equal_, and get_equal_name_.

a) check_star_ is a write-around which calls the new check_star_name_ program, which will be installed in system 20.13 (already approved and submitted).

b) equal_ is a write-around which calls get_equal_name_.

c) get_equal_name implements the extended equals convention, providing a more standard calling sequence than equal_.

Implications: Refer to MSB-103

ersion 2
30919

MULTICS CHANGE REQUEST	MCR 054 Page 1 of 1 Pages

TITLE: Bug fix to 355 Software

AUTHOR: R. B. Snyder

SOURCE: (if external) e.g., "User", "Marketing"

local

STATUS	DATE
Written	9-25-73
Approved X	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR

Implemented in System 20.11k

Objections/Comments:

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

REASONS:

A bug which was crashing the system was fixed and installed an an emergency fix in system 20.11K.

Summary: A bug arose wherein it became possible for status words from the 355 to be delayed in the 6180 mailboxes. When a stop_channel function was executed by the 6180 (a function which, among othere things, throws away all queued status for a given tty channel), status was thrown away in the 355 but not in the 6180 mailboxes.

Implications: This bug has been fixed and installed as a fait accompli. This MCR is merely being submitted after the fact to complete the documentation on the installation.

Versi-n 2
730919

MULTICS CHANGE REQUEST

MCR 055
Page 1 of 2 Pages

TITLE: New Stack and Fault Information Routines

AUTHOR: Melanie B. Weaver

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/28/73
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement	X	Simplification
	Reliability Improvement	X	Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

Should be documented in SWG.

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

REASONS:

There is a need for several new stack and fault information routines. First, there is much information in the stack and/or machine conditions of common interest to several subsystems, namely debug, trace_stack and default_error_handler_ but which needs some interpretation to be useful. Second, it would be helpful to the user if there was more information easily available about software conditions as there currently is for hardware conditions. Third, it would be more compatible for condition handlers to obtain their input information via a subroutine call rather than from an argument list. The size of the condition argument list cannot be changed any more and there already is desirable information which is not directly available from it. Use of subroutines will also enable pl1 on-units to obtain all condition information.

Installation of these routines should make possible more effective and common use of pl1 language constructs so that eventually the analogous Multics constructs will be needed much less frequently.

Summary: Several routines are included in this proposal. They are listed below.

find_condition_info_: is given a pointer to a stack frame being used when a condition occurred and returns information relevant to that condition.

continue_to_signal: used by on-units to tell signal_to continue searching the stack for another condition handler after the on-unit returns.

find_condition frame_: returns a pointer to the stack frame associated with the most recent condition to occur before a specified stack frame.

get_entry_name_: is given a pointer to an external symbol or entrypoint in a segment and returns the associated name. Some validation is performed.

get_def_name_: searches definitions for one whose value matches a given offset. If there is a match, the external sysmol name is returned. This will be a write-around for has_$get_defname_ when the latter is working.

get_link_ptr_: is given a pointer to a text location and tries to find an external reference at or just before the location.

get_link_entry name_: is given a pointer to an entry sequence in the linkage section or to a segdef in a non-object segment and returns the associated name.

is_condition_frame_: checks to see whether a specified stack frame was in use when a condition occurred.

interpret_op_ptr_: examines a location to see if it is in one of the pl1 operator segments. If it is, information is returned concerning the transfer to the operator segment.

stack_frame_exit_: returns the address of the most recent location executed by the owner of a specified stack frame.

find_operator_name_: is given a text reference to one of the pl1 operator segments and returns the name of the operator referenced.

find_ls_owner_: is given a pointer to an active linkage section and returns the segment number of the owning procedure.

is_cls_: checks to see if a pointer is pointing to an active linkage section. This is more difficult when the linkage section is in the stack segment.

find_pathname_: is given a pointer and returns information about the associated pathname, bound segment component, offsets, etc.

interpret_bind_map_: is not a new procedure. The entry compare_offsets, however, is fixed to work with an unbound segment.

These routines have all been implemented, although some still need minor modifications.

Implications: These are all new procedures except for the bug fix to interpret_bind_map_ & should pose no compatibility problems. When these are installed, users should be encourage to use pl1 on-units rather than Multics condition handlers. Versions of trace_stack and debug which use these procedures should be installed (they have also already been largely implemented).

These procedures should not be installed until hcs_$get_defname_is working and until the new stack frame flags are being used.

Version 2
7009195

MULTICS CHANGE REQUEST	MCR 056 Page 1 of 2 Pages

TITLE: Changes to default system condition handler and signaling of pl1 conditions

AUTHOR: Melanie B. Weaver

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/28/73
Approved ✓	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement	X	Simplification
	Reliability Improvement	X	Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: The default system condition handler does not adequately handle software signalled conditions, including pl1 defined conditions. Also, the latter currently are handled completely differently from other conditions. Messages might be more informative if the support bit in the stack frame were interpreted(when it is inplemented). The code in default_error_handler would be clearer and easier to maintain if use were made of the new stack and fault information routines.

Summary: Parts of default_error_handler_, particularly the sections concerned with obtaining and formatting names will be rewritten to use the new stack and fault information routines. This will enable procedure names for software conditions and for non-support frames to be printed. Several changes will be made to the code that formats messages in general. Procedures that signal pl1 conditions will be changed to use a standard info structure; this can be done in a few central locations. default_error_handler will be changed to interpret these pl1 info structures. The procedures that obtain ondata for the pl1 builtin functions will also be changed to use the info structures. The code to process trap before link and trap at first reference will be removed from linkage_error_. If a routine is provided to obtain the source statement corresponding to a specified location, the source statement, when available, will be printed in the message.

Implications: With these changes, the system conditon handler will be easier to maintain and will print better messages for more conditions. In particular, the signalling and handling of pl1 conditions should be much simplified.

No interfaces available to users should be changed, except for the disappearance of pl1_signal_. The handler for the area condition will no longer call the Cambridge Project's special area handler. (This is now done if pl1_signal_is called directly for area.)

Detailed Proposal: Some of the changes are already being implemented. Modules being changed include (the list may not be complete):

default_error_handler_
get_ppr_
get_tpr_
interpret_info_struc_
special messages_
get_simple_names_
linkage error_

A few modules will be deleted and some may be added.

The procedures that call pl1_signal_   be changed to call signal_ with the appropriate info structure. (These cannot actually be be installed until signal is changed - but that is the subject of another MCR. Also it will be necessary for the support bit (a stack frame flag) to be used in order to have useful messages.) The work of default_handler_for_pl1_ will be moved to default_error_handler_ modules.

Some of the modules involved are on the system tape and some are in the on-line libraries. To avoid the need for simultaneous hardcore and on-line installations, the following order is proposed. The new default_error handler_(on system tape) will be installed and will be called for pl1 conditions whenever there is a proper info structure. For ondata to behave properly, all pl1 conditions must be signalled in the same manner. To maintain compatibility when the signalling method is changed (on-line) a new version of pll_signal_ (on-line) will be installed which will turn old calls from pll_operators_ (on system tape) into new style calls to signal_. After that, pll_operators_ can be changed to call signal_ directly.

The total time to implement this will probably be 6 to 8 weeks.

Version 2
730919·

MULTICS CHANGE REQUEST

MCR 057
Page 1 of 1 Pages

TITLE: Unlocking bug in the IMP-DIM

AUTHOR: R. K. Kanodia

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9-1473
Approved	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System 20.11g

Objections/Comments:

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

REASONS:

Unlocking bug in the IMP-DIM

Justification: This bug causes the processes using the network to keep waiting in ring_zero for an event that never happens. Eventually the answering service hangs up and Multics has to be shutdown.

Summary: fix imp_input_processor.pl1 to use stacq.

'ersion 2
'30919

MULTICS CHANGE REQUEST

MCR 058
Page 1 of 1 Pages

TITLE: online_dump to process 256K segments

AUTHOR: R. Mullen

SOURCE: (if external) e.g., "User", "Marketing"

STATUS		DATE
Written	✓	9/29/73
Approved	✓	10/04/73
Rejected		
Postponed		
Withdrawn		
Expires		

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
X	Reliability Improvement	X	Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: Components online_dump $ copy_dump_seg_ in bound_od_ (TOOLS) must be prepared in advance for the day when (hardcore) copy_fdump begins outputting 256K segments to >dumps.

SUMMARY: These programs will call hcs_$mas_length_seg on the first of the dump image segments (for the given dump) in >dumps. The returned length will be used instead of 64K.

IMPLICATIONS: None.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 059
Page 1 of 1 Pages

TITLE: bug fix to get_seg_ptr_

AUTHOR: R. Mullen

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/29/73
Approved ✓	10/04/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

Tools library only.

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

REASONS: entry get_seg_ptr_arg_ does not work (TOOLS)

SUMMARY: The program was not picking up stack_frame.prev_sp correctly to get back to its caller's frame's arg_ptr. It returns error_table_$noarg every time.

IMPLICATIONS: None

DETAILED PROPOSAL: I have a working version which has been tested.

Version 2
730919

MULTICS CHANGE REQUEST	MCR 060 Page 1 of 1 Pages

TITLE: Solution to hangup problem

AUTHOR: R. B. Snyder

SOURCE: (if external) e.g., "User", "Marketing"

local

STATUS	DATE
Written	10/3/
Approved	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

	CLASSIFICATION		JUSTIFICATION
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: In certain cases, hangups are not being noticed by the system due to a bug in tty-ctl.

SUMMARY: tty-ctl will be fixed to request status to always be returned when it has a hangup. It was lack of this status which caused hangups to sometimes go unnoticed.

IMPLICATIONS: none

Version 2
730919

MULTICS CHANGE REQUEST

MCR 061
Page 1 of 1 Pages

TITLE: scs_init to detect multiple interrupt cell assignments.

AUTHOR: R. Mullen

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/1/73
Approved #	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement	X	Simplification
X	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: Multiple assignments of an interrupt cell should be detected during system initialization, in an orderly fashion.

SUMMARY: Interrupts cell assignments are specified in the config deck. scs_init reads the config deck and sets interrupt cells as specified. It is proposed that when it is detected a cell is being assigned a second time the system be crashed via syserr, printing the message:

"scs_init:config deck multiply assigns interrupt cell ¬o"

where ¬o will be the number of the appropriate interrupt cell.

IMPLICATIONS: This message should be added to the list of messages which can appear on the operator's console.

DETAILED PROPOSAL: If the handler for an interrupt cell to be set is already set to any handler other than syserr$syserr_init (a catchall) then syserr will be called as described above. A bit will be set to prevent this test from being made before scs_init has set the handler for all interrupts (temporarily) to syserr$syserr_init.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 062
Page 1 of 1 Pages

TITLE: Install new BOS Loader

AUTHOR: Noel Morris

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/1
Approved	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change	X	Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
X	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: This new loader is the loader to be used with the MTS-500 tapes. It is compatible with the current MTS-400 tapes.

SUMMARY: See attached MOSN.

IMPLICATIONS: LOADDM in BOS will have to be modified first to boot from a drive other than 0. (Another MCR.)

COMMENT: This loader (since it is on cards) is written as a GMAP program (ALM doesn't produce binary card images). There is probably no official way to install a GMAP program

Version 2
730919

MULTICS CHANGE REQUEST

MCR 063
Page 1 of 1 Pages

TITLE: Allow BOS to be loaded from tape handler other than drive 0.

AUTHOR: Noel Morris

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	9/30/73
Approved	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction	X	Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: When the MTS-500 tapes are installed, there will be no drive 0. Currently, the program LOADDM in BOS will boot only from handler 0.

SUMMARY: A two line coding change is needed in LOADDM to pick up the device number from the PCW used by the BOS loader (loaded from the card reader).

IMPLICATIONS: BOS will continue to be loadable from drive 0 until the new MTS-500 BOS loader is installed (Another MCR). At that time, an MOSN will be issued.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 065
Page 1 of 1 Pages

TITLE: Fix reused address bug

AUTHOR: Steve Webber

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/1/
Approved #	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System 20.12 a

Objections/Comments:

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

REASONS: Fix "reused address" bug, as well as a few others that showed up while looking for the reused address bug.

SUMMARY: Simple changes to pc, page_fault.

IMPLICATIONS: none - already installed

Version 2
730919

MULTICS CHANGE REQUEST

MCR 066
Page 1 of 1 Pages

TITLE: Remove "hcs_$reset_working_set" function from the system

AUTHOR: Steve Webber

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/2/73
Approved #	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
X	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
X	Performance Improvement	X	Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ______

Implemented in System ______

Objections/Comments:

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

REASONS: The entry "hcs_$reset_working_set" was originally designed to be used in conjunction with the pre-paging algorithm. It turns off the "used" bits in ptw's for some pages which were used by the calling process since the process was last loaded.

SUMMARY: The target of the gate should be removed and the gate itself should be changed to a simple return. Eventually, the gate should be removed altogether.

IMPLICATIONS: Users should be warend that the entry is obsolete and that it will go away some day.

ersion 2
30919

MULTICS CHANGE REQUEST

MCR 067
Page 1 of 6 Pages

TITLE: Add "mexp" to system--a macro expanding preprocessor for alm

AUTHOR: Steve Webber

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/1/73
Approved	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

Install in Tools, no user documentation.

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

REASONS: The current hardcore gate segments are maintained with the uninstalled tool mexp. The program is thought to be of enough general use to be placed in the Tools library.

SUMMARY: mexp is a simple test manipulative macro expanding preprocessor designed to be used in conjunction with alm. It provides for

1) macro expansion with argument substitution
2) iteration (over macro arguments) when expanding
3) the dup, ife, and ine pseudo-ops
4) unique symbol generation

IMPLICATIONS: None, other than parts of the system may develop a dependency on such a macro possibility.

DETAILED PROPOSAL: See attached writeup.

Name: macro_expand, mexp

The macro_expand command is a fairly simple text manipulative program which is to be used in conjunction with the ALM assembler. The program takes "mexp" source segments, expands any macros found therein, and generates as output an expanded text file suitable as input to the ALM assembler.

The macro_expand command is purely text manipulative and does not have the capability for doing any expand time decision making other than comparison of character strings. Conditional expansion of "code" is possible with the use of ine and ife pseudo-operations. In addition the ability to generate unique symbols within macros is provided. A limited form of iteration is also provided which allows for repetitive expansion of macro components. A macro must be defined before it is used as the macro expander makes only one pass over the input text.

Usage

```
                macro_expand name
```

The program will expand name.mexp (or name if name ends in .mexp) and will generate as output name.alm.

Notes

The format of a "mexp" source program is quite similar to an ALM source program. The main difference is that macro definition and macro expansion statements are interspersed with the normal ALM statements. To define a macro the "&macro" pseudo-op is used. The format of this is as follows:

```
             &macro          macro-name

             macro-body

             &end
```

If the string "&macro" is found in the context of an ALM opcode or pseudo-op it is interpreted as the start of a macro definition.

The name of the macro is the next "word" on the line. The body of the macro is all of the text up to but not including the next matching "&end" in the source text. The body of the macro may include any text which when expanded by the rules specified below will yield valid ALM source code.

The following control sequences direct the macro expander to act in a special way:

1) &0, &1, &2, ... the character "&" followed immediately by any decimal integer (< 100) is replaced, upon expansion, with the corresponding argument passed to the macro (see below).

2) &u is expanded to be a unique character string which is different from any other such strings expanded. The string will be 8 characters long.

3) &U is expanded to be a unique character string. However multiple occurrences of &U within the same macro will yield the same string.

4) &p is expanded to be the 8 character string of the previous &u expansion.

5) &n is expanded to be the 8 character string of the next &u expansion.

6) &(n indicates the beginning of an "iteration" sequence. The text following the &(n and up to but not including the next &) will be expanded at run time only if there are additional parameters to the macro which have not been "used up". (See below.)

7) ife (ine) if ife or ine occur in the context of an opcode or it causes conditional expansion of the text up to the next matching "ifend" depending on the equality (inequality) of the first two parameters to the pseudo-op. The equality comparison is strictly a character string compare.

8) dup causes the text up to the matching "dupend" to be duplicated n times where n is the decimal value of the (first) parameter to the pseudo-op.

9) &i is expanded to be the particular parameter in an iterated list for which the current iteration expansion is being done. (see below.)

10) &x is expanded into the decimal integer correspondir to the argument position of the iteration argument for which the current iteration expansion is being done. (see below.)

Notes

If a parameter is not specified for a particular parameter position a zero length string will be used for expansion.

The argument "&0" expands to be the first label on the statement invoking a macro.
Any parentheses around a parameter will be stripped off upon expansion. Parentheses used in this manner are treated as quoting characters.

Blanks may not appear in a macro parameter list unless within a parenthesized parameter.

The unique identifiers generated by mexp are of the form

```
                    ...00000
                    ...00001
                       .
                       .
                       .
```

for &u, &p and &n expansions, and

```
                    ..!00000
                    ..!00001
                       .
                       .
                       .
```

for &U expansions.

The iteration feature is invoked by passing a parenthesized list of parameters in the parameter position for the specified iteration. The parameter number for an iteration sequence immediately follows the "&(" of its definition.* Iterated arguments are scanned in the same manner as macro arguments and hence quoting may be done with parentheses.

The pseudo-operation "&macros" can also be used to define macros. When this pseudo-op is encountered, its parameter is treated as pathname of a macro definition file. The macros contained in the specified file are defined in the same way as if the macro definitions were in the text directly. If no argument is given to the "&macros" pseudo-op or the argument is "system" the normal system macro file will be used.

Examples

The following macro definition shows a typical expansion:

```
        &macro      load
                    ld&1       &2
        &end
```

*If no parameter number is specified "1" is assumed.

With this macro definition the following expansions might result:

```
          load      x0,temp   -->  ldx0      temp

          load      a,(sp|3,*) --> lda       sp|3,*
```

Note the use of parentheses in the second expansion to cause the comma to be ignored as a parameter delimiter.

The following macro shows a typical use of a unique symbol:

```
         &macro     test
         &U:        lda        &l
                    tnz        &U
                    sta        &2
         &end
```

This macro might expand as follows:

```
         test   a,b         -->          ..!00000:    lda   a
         test   c,d                                   tnz   ..!00000
                                                      sta   b
                                         ..!00001:    lda   c
                                                      tnz   ..!00001
                                                      sta   d
```

The following example shows how iteration might be used:

```
         &macro     table
         &(l        vfd        18/&i,18/&0
         &)
         &end
 e1:     table     (4,6,8,10)      -->       vfd       18/4,18/e1
                                             vfd       18/6,18/e1
                                             vfd       18/8,18/e1
                                             vfd       18/10,18,e1
```

The following example shows how conditional expansion may be used:

```
         &macro     meter
                    lda        &l
                    ife        &2,on
                    aos        meterword,al
         &end       ifend
                    meter      foo,on     -->   lda    foo
                                                aos    meter_word,al
```

The following example shows how &x might be used:

```
&macro outer
       lda  &i, dl
       sta  temp + &x + 1

&end   outer  (1,2,3,4) → lda  1, dl
                          sta  temp +1-1
                          lda  2, dl
                          sta  temp +2-1
                          lda  3, dl
                          sta  temp +3-1
                          lda  4, dl
                          sta  temp +4-1
```

ersion 2
30919

MULTICS CHANGE REQUEST

MCR 068
Page 1 of 1 Pages

TITLE: fix msa_manager_'s setting of MSF indicator

AUTHOR: Gary C. Dixon

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/3/73
Approved #	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
X	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ______________

Implemented in System ______________

Objections/Comments:

Approval of this MCR does not imply approval of msa_manager_.

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

REASONS: msa_manager_ should set the MSF indicator of the msa's (multi-segment area' directory in a manner consistent with msf_manager_'s setting on msf directories.

SUMMARY: msa_manager_ is now incorrectly setting the MSF indicator of the msa directory to a value of 1 less than the number of component segments. It should be set to the number of component segments. I propose to implement this bug fix, on the grounds of cleanliness. (It is a one-line fix.)

IMPLICATIONS: None. No programs currently use the MSF indicator of an msa.

Version 2
730919

MULTICS CHANGE REQUEST	MCR 069 Page 1 of 1 Pages

TITLE: **changes to signalling and unwinding**

AUTHOR: **M. Weaver**

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/4/73
Approved #	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension	X	Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement	X	Simplification
	Reliability Improvement	X	Generalization
			Bug Fix

Replaced by proposal MCR ______

Implemented in System ______

Objections/Comments:

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

REASONS: Some changes need to be made to the signalling mechanism in order to implement the PL/I - defined condition mechanism properly. This is a good time to eliminate the special on units for cleanup and default handlers, replacing them with ordinary on units for cleanup and any_other. Some improvements can be made in error processing.

SUMMARY: Change signal_ to:
recognize the new PL/I condition information structure so that the special entry may be phased out,
handle snap and system for PL/I,
call a default handler before crawling out to see if the condition can be handled and execution resumed,
copy information structures on crawlouts,
copy wall crossing conditions on crawlouts when the current ring was entered via a fault
change the unwinder to:
recognize and signal unwinder_error (done on the 645 but not the 6180),
signal cleanup.

Implementation time should be about 2 weeks, plus some development machine time.

IMPLICATIONS: These changes should be transparent to users except for the occasional extra information available on crawlouts. Users have been told that all information structures must have a standard header by Oct. 31, so the number of words to copy should always be available.

ersion 2
30919

MULTICS CHANGE REQUEST	MCR 070 Page 1 of 1 Pages

TITLE: fix send_message * proj bug

AUTHOR: R. Mullen

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/4/73
Approved	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
X	Reliability Improvement		Generalization
		X	Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS:

The command lines:

```
                    send_message * proj message words
or                  send_message pers * message words
```

give the user an IPR fault

SUMMARY: This occurs because a begin block does not have a closing end statement.

IMPLICATIONS: Should be installed in special session or when no users are logged in, lest one accepts messages with one version and later tries to send messages with the other.

DETAILED PROPOSAL: I have a working copy.

Version 2
730919

MULTICS CHANGE REQUEST

MCR 071
Page 1 of 1 Pages

TITLE: **Print detailed status on disk errors**

AUTHOR: **N. I. Morris**

SOURCE: (if external) e.g., "User", "Marketing"

FED Requested

STATUS	DATE
Written	10/4/73
Approved #	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
	Extension		Conformance to Standard
	Restriction		Increased Consistency
	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: **The detailed status held in a disk drive (DSU-190, DSU-181) is often needed by FED when an error occurs.**

SUMMARY: **The disk DIM will be modified to do an RSR command whenever an error occurs. The 9 bytes of resulting detailed status will be printed via syserr.**

ersion 2
30919

MULTICS CHANGE REQUEST	MCR 072 Page 1 of 10 Pages

TITLE: Change Multics standard tape format record length

AUTHOR: N. Morris

SOURCE: (if external) e.g., "User", "Marketing"

STATUS	DATE
Written	10/5/73
Approved	10/9/73
Rejected	
Postponed	
Withdrawn	
Expires	

CLASSIFICATION		JUSTIFICATION	
	Incompatible Change		Marketing Requirement
X	Extension		Conformance to Standard
	Restriction		Increased Consistency
X	Performance Improvement		Simplification
	Reliability Improvement		Generalization
			Bug Fix

Replaced by proposal MCR ____________

Implemented in System ____________

Objections/Comments:

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

REASONS: It is proposed that the record length of tape records in Multics standard tape be changed from 272 to 1040 words. See the attached MTB for arguments. See also attached MPM section.

SUMMARY: Tape DCM's, DIM's and DSM will have to be changed to allow for reading either size record and writing 1040 records.

IMPLICATIONS: An old copy of the reloader and dumper must be preserved so that tapes can be created for other sites.

TO: DISTRIBUTION

FROM: N.I. MORRIS

DATE:

SUBJECT: MULTICS STANDARD TAPE RECORD LENGTH

Problems with Current MST Record Length

With the current 272 word MST record, a large percentage of tape consists of inter-record gap. This causes a large amount of the usable surface on a tape to be wasted. It also places a limitation on the effective data transfer rate from an MST in that a large percentage of time is spent in moving the tape over inter-record gaps. With the installation of 1600 bpi tape handlers, and the future possibility of having 6250 bpi handlers, these problems worsen. It would be advantageous, from the point of view of more fully utilizing the space on a tape and increasing the effective data transfer rate, to increase the size of the MST record.

Table 1 summarizes the parameters associated with the current 272 word MST record. It is obvious that with 1600 bpi tapes, almost half of the tape is wasted. Also, a significant increase in data rate does not occur. (Note that MTS-500 handlers are actually slower than MTS-400 handlers.)

Physical Considerations in Choosing a Tape Record Length

Magnetic tape records may be made arbitrarily long. However, tape is an imperfect medium, subject to physical abuse. The tape surface may deteriorate and deform. The tape edges may become crimped through mishandling. Error rates become greater at the ends of the tape. The leader experiences a great deal of wear from loading and unloading. The end of the tape undergoes some deformation from the presence of the EOT reflector on a tightly wrapped reel of tape.

It is obvious that the longer a tape record, the greater the possibility of an error in that record. It can also be seen that some errors may occur once per revolution of the tape reel. Thus, one limiting factor for tape record length is that a record should be shorter than the innermost circumference of a tape reel. (This is $5.125 \, X \pi \simeq 16$ inches.)[1] Studies done by Honeywell in Oklahoma City indicate that a record length of 3 to 4 inches is optimal.

[1] ANSI X3B1/402

Software Considerations in Increasing MST Record Length

Increasing the size of the MST tape record will affect both the tape DCM and the tape DSM. The wired-down buffers in the DCM will have to increase in size and the buffering strategy of the DSM will require minor modification. The number of tape records which can be read or written in a single call to the tape DCM (currently 6) will no doubt have to be decreased in order to conserve wired-down core. As many records as possible, though, should be transferred in a single DCM call in order to minimize system overhead in processing interrupts, waits and notifies, etc.

One user of the tape DSM which is somewhat sensitive to tape record length is the dumper/reloader. The backup system always writes 256 word logical records on tape, padding shorter records to 256 words. It uses this fact to recover from tape read errors and resynch itself with the physical tape. If the increased record length is not a multiple of 256, the backup system might experience grave problems in reloading a tape with bad spots on it.

New MST Record Length

On the basis of the above information, a new MST record length of 1040 words is proposed. This will consist of an eight word record header, a 1024 word record body, and an eight word record trailer. The parameters associated with such an MST record are found in Table 2. MPM Reference Guide Section 5.3 has been rewritten to conform to the new record length and may be found in Appendix 1.

Note that, in all cases, the physical record length on tape is less than 16 inches, and for 1600 bpi tapes it is close to the optimal values discussed above. Note, too, that less than 20 percent of the tape is wasted, and that with 1600 bpi tapes, a significant increase in effective data rate results.

Tape System	Track/ Density	Tape Speed	Frames in MST Record	Frames of Data	MST Record Length	Data Length	Gap Length	Effective Date Rate	Present Use
MTS-400	7 track 800 bpi	150 ips	1632	1536	2.04"	1.92"	.75"	13763 wds/sec	68.8%
	9 track 800 bpi	150 ips	1224	1152	1.53"	1.44"	.6"	18028 wds/sec	67.6%
MTS-500	7 track 800 bpi	125 ips	1632	1536	2.04"	1.92"	.75"	11470 wds/sec	68.8%
	9 track 800 bpi	125 ips	1224	1152	1.53"	1.44"	.6"	15023 wds/sec	67.6%
	9 track 1600 bpi	125 ips	1224	1152	0.765"	0.72"	.6"	23357 wds/sec	52.6%

TABLE 1: 272 WORD MST RECORD

Tape System	Track/ Density	Tape Speed	Frames in MST Record	Frames of Data	MST Record Length	Data Length	Gap Length	Effective Data Rate	Present Use
MTS-400	7 track 800 bpi	150 ips	6240	6144	7.8"	7.68"	.75"	17964 wds/sec	89.8%
	9 track 800 bpi	150 ips	4680	4608	5.85"	5.76"	.6"	23814 wds/sec	89.3%
MTS-500	7 track 800 bpi	125 ips	6240	6144	7.8"	7.68"	.75"	14971 wds/sec	89.8%
	9 track 800 bpi	125 ips	4680	4608	5.85"	5.76"	.6"	19845 wds/sec	89.3%
	9 track 1600 bpi	125 ips	4680	4608	2.925"	2.88"	.6"	36312 wds/sec	81.7%

TABLE 2: 1040 WORD MST RECORD

MULTICS STANDARD MAGNETIC TAPE FORMAT

This section describes the standard physical format to be used on seven track and nine track magnetic tapes on Multics. Any magnetic tape not written in the standard format described here is not a Multics standard tape.

Standard Tape Format

The first record on the tape following the beginning of tape (BOT) mark will be the tape label record. Following the tape label record will be an end of file (EOF) mark. Subsequent reels of a multireel sequence will also have a tape label record followed by an EOF mark. (An EOF mark is the standard sequence of bits on a tape which is recognized as an end of file by the hardware.)

Following the tape label record and its associated EOF mark are the data records. An EOF mark will be written after every 128 data records with the objective of increasing the reliability and efficiency of reading and positioning within a logical tape. Records which are repeated because of transmission, parity, or other data alerts are not included in the count of 128 records. These 128 record groupings are referred to below as physical files.

An end of reel sequence will be written at the end of recorded data. An end of reel sequence is:

EOF mark

end of reel record

EOF mark

EOF mark

Standard Record Format

Each physical record consists of a 1024 word (36864 bit) data space enclosed by an eight word header and an eight word trailer. The total record length is then 1040 words (37440 bits). The header and trailer are each 288 bits. This physical record will require 4680 frames on nine track tape and 6240 frames on seven track tape. This is approximately 5.85 inches on nine track tape and 7.8 inches on seven track tape, at 800 bpi not including interrecord gaps. (Record gaps on nine track tapes are approximately 0.6 inches and on seven track tapes, approximately 0.75 inches, at 800 bpi.)

For 1600 bpi nine track tape, the record length is approximately 2.925 inches (with an inter-record gap of approximately 0.5 inches).

Physical Record Header

The following is the format of the physical record header:

Word 1: Constant with octal representation 670314355245.

Words 2 and 3: Multics standard unique identifier (70 bits, left justified). Each record will have a different unique identifier. The fact that the records' unique identifiers are sequential can be used to detect the end of relevant data on a tape when no end of reel record was written.

Word 4: Bits 0-17: the number of this physical record in this physical file, beginning with record 0. (The first record following an EOF mark will have a physical record count of 0.)

Bits 18-35: the number of this physical file on this physical reel, beginning with file 0.

Word 5: Bits 0-17: the number of data bits in the data space, not including padding.

Bits 18-35: the total number of bits in the data space.

Word 6: Flags indicating the type of record. Bits are assigned considering the left most bit to be bit 0 and the right most bit to be bit 35. Word 6 also contains a count of the rewrite attempt, if any.

Bit	Meaning
0	If 1, this is an administrative record (one of bits 1 through 13 is 1).
1	If 1, this is a tape label record.
2	If 1, this is an end of reel record.
3-13	Reserved and must be zero.

14 If 1, one or more of bits 15-26 are set.

15 If 1, this record is a rewritten record.

16 If 1, part or all of the record is filled in with the padding bit pattern (see word 5 of the record trailer description).

17 If 1, this record was written following a hardware end of tape (EOT) condition.

18 If 1, this record was written synchronously; that is, control did not return to the caller until the record was written out.

19 If 1, the logical tape continues on another reel. (This bit is defined only for an end of reel record.)

20-26 Reserved and must be zero

27-35 If bits 14 and 15 are 1, this quantity indicates the number of the attempt to rewrite this record. If bit 15 is 0, then this quantity must be 0.

Word 7: Contains the checksum of the header and trailer excluding word 7, i.e., excluding the checksum word. (See the MPM Reference Guide section on Standard Checksum for a description of standard checksum computation.)

Word 8: Constant with octal representation 512556146073.

new page

Physical Record Trailer

The following is the format of the trailer:

Word 1: Constant with octal representation 107463422532.

Words 2 and 3: Standard Multics unique identifier (duplicate of header).

Word 4: Total accumulative number of data bits for this logical tape (not including padding and administrative records).

Word 5: Padding bit pattern (its use is described below).

Word 6: Bits 0-11: reel sequence number (multireel number), beginning with reel 0.

Bits 12-35: physical file number, beginning with physical file 0 of reel 0.

Word 7: The number of the physical record for this logical tape, beginning with record 0.

Word 8: Constant with octal representation 265221631704.

Note: The octal constants listed above were chosen to form elements of a single error correcting code whether read as eight bit tape characters (nine track tape) or as six bit tape characters (seven track tape).

Administrative Records

The standard tape format includes two types of administrative records: 1) a tape label record; and 2) an end of reel record.

The administrative records are of standard length: eight word header, 1024 word data area, and eight word trailer.

The tape label record is written in the standard record format. The data space of the tape label record contains:

Words 1-8: 32 character ASCII installation code. This identifies the installation which labelled the tape.

Words 9-16: 32 character ASCII reel identification. This is the reel identification by which the operator stores and retrieves the tape.

The remaining words are a padding pattern.

The end of reel record contains only padding bits in its data space. The standard record header of the end of reel record contains the information which identifies it as an end of reel record. (Word 6, bits 0 and 2 are 1.)

Density and Parity

Both nine track and seven track standard tapes will be recorded in binary mode with odd ones having lateral parity. Standard densities are 800 frames per inch (bpi) (recorded in MRZI mode) and 1600 bpi (recorded in PE mode).

Data Padding

The padding bit pattern will be used to fill administrative records and the last data record of a reel sequence.

Write Error Recovery

Multics standard tape error recovery procedures differ from the past standard techniques in that no attempt is made to backspace the tape on write errors. If a data alert occurs while writing a record, that record will be rewritten. If an error occurs while rewriting the record, that record will again be rewritten. A reasonable number of attempts may be made to write the record. No backspace record is issued.

The above write error recovery procedure is to be applied to both administrative records and data records.

Compatibility Consideration

Software shall be capable of reading Multics Standard Tapes which are written with records with less than 1024 words in their data space. In particular, a previous Multics Standard Tape format specified a 256 word (9216 bit) data space in a tape record.

(END)