

TO: Distribution
FROM: Joan Scott
DATE: 12 December 75
RE: Multics Change Requests

Enclosed are copies of Multics Change Requests which were approved from 1 December 75 through 15 December 75.

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

Multics Change Request

TITLE: Implement mailbox_set_max_length command		STATUS	DATE
AUTHOR: S. Herbst		Written	10/20/75
<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 type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	P 11/04/75 A 12/02/75
		Expires	06/02/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	AG92
		PLMS (AN #)	AN69
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Install mailbox_set_max_length command and entry point mailbox_\$set_max_length_index.

REASONS:

When an existing security bug has been fixed, users will be unable to set the max length on their ring 1 mailboxes. Max length is useful to prevent long mail from using up your quota.

DETAILED PROPOSAL:

The command implemented in the module ms_create calls new gate entry mailbox_\$set_max_length_index which in turn calls the new entry point mbx_mseg_\$set_max_length to do the work in ring 1.

Entry: mailbox_\$set_max_length_index

This entry point sets the maximum length of a mailbox.

Usage: dcl mailbox_\$set_max_length_index entry

(fixed bin (17),fixed bin(18), fixed bin(35));

call mailbox_\$set_max_length_index

(mbx_index, length, code);

where:

1. mbx_index is the index of a mailbox. (Input)
2. length is the desired maximum length.
If this number is not a multiple of 1024 words,
it is rounded up. This number must not be
less than the current length of the mailbox.
(Input)
3. code is a standard status code. (Output)

Name: mailbox_set_max_length, mbsml'

This command sets the maximum length of a mailbox.

Usage: mbsml path length -control_args-

where:

1. path is the pathname of a mailbox. If the suffix mbx is missing, it is assumed. The star convention is allowed.
2. length is the maximum length. If this number is not a multiple of 1024 words, it is rounded up with a warning.
3. -control_args can be:
 - decimal, -dc length is a decimal number. (This is the default.)
 - octal, -oc length is an octal number.
 - brief, -bf suppresses the warning that length has been rounded to the next higher multiple of 1024 words.

The new maximum length must not be less than the current length of the mailbox.

Multics Change Request

TITLE: Install create_data_segment_ as a tool.		STATUS	DATE
AUTHOR: Bernard Greenberg		Written	11/11/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL		Status	P 11/18/75 A 12/02/75
-Planned for System MR <u>4</u>		Expires	06/02/76
-Fixes Bug Number(s) _____		DOCUMENTATION CHANGES	
-Documented in MTB attached		Document	Specify One or More
-User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no		MPM (Vol, Sect.)	
-Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		PLMS (AN #)	51
-Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse		MOSN (Sect.)	
-Replaces MCR _____		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: Install create_data_segment_ as a tool for creating data bases which are now constructed by ALM.

Reasons: ALM is a sloppy and inelegant tool for constructing data bases with segdefs. create_data_segment_ allows full PL/1 language specification of data segment structure. Since PL/1 structures are used to define data base layouts, the benefits of include files for defining these layouts may be gained.

The prelinking and area management projects have both made use of this subroutine as a private tool and found it valuable.

Implications: One of the major reasons for the use of ALM will be eliminated. The viability of this interface as a user tool will be able to be determined.

Detailed proposal. Install create_data_segment_ in tools. A sample MPM writeup on data segments and data segment creation is attached.

NOTES: create_data_segment_ produces fully standard object segments, with optional separate static. ALM is currently the only method available for constructing data segments.

create_data_segment_ is a tool for constructing data segments: it is NOT intended to be a language for data base specification.

DATA SEGMENTS

A data segment is a segment referenced symbolically by Multics programs, which contains data other than executable code. Data segments may be referenced by either of two ways: they can be made known via calls to the subroutines `hcs_$initiate` and `hcs_$initiate_count` (See MPM Sec. 999), in which case they may be addressed via the pointers returned by these two subroutines, or they may be addressed symbolically as external data in a PL/I or Fortran program.

In the case of a data segment which is explicitly made known by programs, the data contained in it may have any structure desired by the writer of the programs. Such a segment can be created by the `create` command (See MPM Sec. 999). Below is a sample PL/I program designed to add two integers in a data segment called "my_data", and store the result in a reserved location in that segment.

```
adder: procedure;
    declare p pointer;          /* Pointer to segment */
    declare hcs_$initiate entry (character (*), character (*),
        character (*), fixed binary, fixed binary, pointer,
        fixed binary (35));
    declare 1 my_data_structure based, /* Layout of data segment
        2 first_number fixed binary,
        2 second_number fixed binary,
        2 answer fixed binary;

    declare code fixed binary (35);

    call hcs_$initiate (">udd>Washington>States", "my_data", "",
        0, 0, p, code);        /* Make the segment known */

    p -> my_data_structure.answer =
        p -> my_data_structure.first_number +
        p -> my_data_structure.second_number;

end;
```

In this program, the pointer `p` is set by `hcs_$initiate` to point to the segment `>udd>Washington>States>my_data`. All references via the pointer `p`, such as the assignment statement above, are actually direct references to that segment. (See "Making a Segment Known", in MPM Part 998). This technique is the preferred way of dealing with a data segment with user-defined structure.

Data segments can be addressed as external data from a PL/I or Fortran program. In this case, the segment will be found via the search rules (see "Search Rules" in "Programming

Environment", MPM 997) at the time the program first references the data. In PL/I, a structure or other variable may be declared as residing in an external segment by declaring it external. The last character of the name of such an object must be "\$", and the rest of the name will be the reference name which will be searched for.

The following PL/I program prints out an array of names and numbers stored in the segment "baseball_data".

```
print_ba: procedure;

  declare sysprint file;
  declare 1 baseball_data$ external,      /* The external segment */
          2 num_players fixed binary,    /* Number of players repr
          2 data (1 refer (baseball_data$.num_players)),
          3 name character (40),
          3 average fixed decimal (3, 3);  /* e.g., .567 */

  declare i fixed binary;                /* loop index */

  do i = 1 to baseball_data$.num_players;
    put file (sysprint) list
      (name (i), " average = ", average (i));
  end;
end print_ba;
```

In the above program, all references to elements of the structure `baseball_data$` will be interpreted as being in the segment `baseball_data`, whose structure is defined by the declaration above. The exact pathname of the segment `baseball_data` will be determined by the dynamic linker, which will search for a segment by this reference name the first time that this program is executed in any given process.

There is one more type of data segment that can be used. A data segment may contain symbolically named structures and variables. In this case, the data segment is a standard Multics object segment which is produced by the `create_data_segment_subroutine` (See below). Such a data segment is called a structured data segment. The data objects in the data segment are referenced individually by name, as external objects in a PL/I program (see below). Such objects are referenced and declared with names such as `my_seg$datum_1` in PL/I, where a dollar sign (\$) appears in the middle of the name, and the object is declared external. The part of the name before the dollar sign is the reference name by which the dynamic linker will search for the segment. The part of the name beyond the dollar sign is called the `segdef` name, and is the name of a particular datum in that segment. This type of data segment allows a great deal of flexibility, because no programs that use the data segment need know its structure, or the

relative layout of data items within it. Since items within the segment are referenced directly by name, and these references are resolved by the dynamic linker instead of the translators, it is very easy to change the layout of such a data segment, adding and subtracting items as necessary.

The following PL/I program excerpt might be part of a report printing program. The name of the company and its president are kept in a structured data segment called `company_data`.

```
declare company_data$company_name char (50) external;
declare company_data$president char (32) external;

...

put file (reportfile) skip
  list (company_data$company_name, ", ",
        company_data$president, ", President.");
```

Note that the program excerpt above does not know the layout of the segment `company_data`. It is cognizant only of the existence of two named objects within it. Such segments are located via the search rules, by the dynamic linker, when the program references the segment for the first time.

CREATION OF STRUCTURED DATA SEGMENTS

In order to create a structured data segment, i.e., a data segment with symbolically named objects within it, the `create_data_segment_subroutine` is used, in the following way.

A PL/I program is written, which defines, via PL/I structures, the layout of the data segment to be produced. One structure may be provided to define objects residing in the impure part of the data segment, and one for the pure part. Statements in the PL/I program place data into these structures, which will be the data to be placed initially in the data segment to be created. The last statement in this program will be a call to the subroutine `create_data_segment_`, providing to it the names, locations, and lengths of these structures, and the name of the data segment to be created. The PL/I program is then compiled and run. When it runs, and calls `create_data_segment_`, a data segment will be created. It will have the name specified and the contents of the structures provided by the program. The names of the second-level structure components of the structures provided will become the `segdef` names of the objects in the data segment corresponding to those components.

Here is a PL/I program which might be used to create the data segment used in the previous example.

```
create_company_data: procedure;

    declare create_data_segment_ entry (ptr, fixed bin, char (*),
                                        ptr, fixed bin, char (*),
                                        ptr, fixed bin, char (*),
                                        char (*), bit (1));

    declare 1 template,                /* Template for new segment */
            2 company_name char (50),
            2 president char (32);

    template.company_name = "Ajax Wax Works, Inc.";
    template.president = "E. L. Gildersleeve, Jr.";
                                /* Specify initial contents */
    call create_data_segment_ (addr (template), size (template), "ter
                                null(), 0, "",
                                null(), 0, "",
                                "company_data", "0"b);
                                /* Now create the data segment */

end;
```

See the writeup of `create_data_segment_` in the MPM, Sec. 996.

create_data_segment_create_data_segment_

The `create_data_segment_` subroutine generates symbolically addressable standard object segments with specified contents and inbound definition names. Such segments can have their contents addressed as `a$b` from a PL/I program.

```
Usage:  declare create_data_segment_ entry
        (ptr, fixed bin, char (*),
         ptr, fixed bin, char (*),
         ptr, fixed bin, char (*),
         char (*), bit (1));

        call create_data_segment_
        (addr (text_template), size (text_template), "text_tem
         addr (link_template), size (link_template), "link_tem
         addr (static_template), size (static_template), "stat
         "segname", "0"b);
```

Where:

`text_template`, `link_template`, and `static_template` represent any three arbitrary level-1 structures whose contents are to be placed in the text, linkage-resident static, and separate static portions of the object segment.

`segname` is the name of the object segment to be constructed, and the "0"b argument is reserved for future use.

The `create_data_segment_` subprogram is used in the following way. A user desiring to create a data segment writes a PL/I program in which he defines PL/I structures for the regions in the text, link, and separate static sections of the object segment to be created. All sections are optional, and specifying the corresponding structure name as "" causes the corresponding section not to be generated. He then compiles and runs this program. The call to `create_data_segment_` will cause a standard Multics object segment to be generated.

The object segment which will be generated, with the name given as `"segname"`, will have the contents of the corresponding structures in the correct sections. Each second-level component name will be used to generate a `segdef` (inbound definition) by that name to the corresponding point in the object segment.

Before calling `create_data_segment_`, the user can initialize his template structures in any way he desires, with either the "initial" attribute or explicit code.

The program which contains the structures, and calls

Multics Programmer's Manual

`create_data_segment_`, must be compiled with the `-table (-tb)` compiler option.

Notes:

Separate static will be generated if and only if a non-null static structure name is given. `create_data_segment_` reports errors via `com_err_`, as it may be considered a specially-called command.

All text and linkage resident information is relocated absolutely; hence, one must be wary of dynamic initialization which creates threads or pointers in data bases which are expected to be bound.

The brief name of this translator is `CompData`.

It is essential that the structures supplied to `create_data_segment_` are referenced in the calling program, or the compiler will not provide symbol table entries.

See the MPM Reference guide section on Data Segments for an example of the use of this program.

TITLE: MCS bug fixes		STATUS	DATE
AUTHOR: M. Grady		Written	19 November 75
-Coded in: <input type="checkbox"/> PL/I <input type="checkbox"/> ALM <input checked="" type="checkbox"/> other- explain in DETAILED PROPOSAL		Status	A 12/02/75
-Planned for System MR 3.1		Expires	06/02/76
-Fixes Bug Number(s) _____		DOCUMENTATION CHANGES	
-Documented in MTB _____		Document	Specify One or More
-User/Operations-visible		MPM (Vol, Sect.)	
Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		PLMS (AN #)	85
-Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		MOSN (Sect.)	
-Performance: <input checked="" 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:

Certain problems have been discovered in MCS.

1. Status and data ICW's will store at the incremented address even after tally runout. This sometimes destroys good data.
2. Excessive interrupts from a particular channel could crash 355.
3. Incomplete output messages for remote printers were not handled properly.

DETAILED PROPOSAL:

Install MCS to fix these problems. Coded in 355map.

Multics Change Request

TITLE: Install 355 utility routines		STATUS	DATE	
AUTHOR: M. Grady		Written	19 November	
-Coded in: <input type="checkbox"/> PL/I <input type="checkbox"/> AIM <input checked="" 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 A 12/02/75	
	<input type="checkbox"/>	Lib. Maint. Tools	Expires 06/02/76	
	<input type="checkbox"/>	Sys. Anal. Tools	DOCUMENTATION CHANGES	
	<input checked="" type="checkbox"/>	Sys. Prog. Tools	Document	Specify One or More
	X 355	BOS	MPM (Vol, Sect.)	
		Salvager	PLMS (AN #) MCS AN85	
		Ring Zero	MOSN (Sect.)	
		Ring One	MPAM (Sect.)	
		SysDaemon/Admin.	MSAM (Sect.)	
		Runtime	Info Segs	
		User Cmmd/Subr.	Other (Name)	
Objections/Comments:			None (Reason)	

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

SUMMARY:

Install updated/corrected source and macro library for the 355 loader and utility routines.

REASONS:

These routines are correct in object form only, and could not be duplicated by assembly from source.

DETAILED PROPOSAL:

Coded in 355map.

TITLE: RCP bug fixes		STATUS	DATE
AUTHOR: B. Silver		Written	21 November 75
<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 18/02/75
		Expires	06/02/76
		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 checked="" type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input 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)	Changes implement defined interface

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

SUMMARY:

Fix bugs in RCP.

REASONS:

Some of these bugs result in RCP not performing the currently defined user/operator interface.

DETAILED PROPOSAL:

1. Fix bug that causes the unassign_resource command to fail to unassign a device that is attached and was explicitly assigned.
2. Fix bug that results in RCP failing to rewind/unload a tape reel that was mounted with the write ring set incorrectly.
3. Change RCP to interpret special status in a way that will help solve the problem of lost special interrupts.
4. Correct error_table_code name errors in assign_resource.

Multics Change Request

TITLE: tape_mult_bug fixes		STATUS	DATE
AUTHOR: B. Silver		Written	21 November
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL		Status	A 12/02/75
-Planned for System MR 3.1		Expires	06/02/76
-Fixes Bug Number(s)		DOCUMENTATION CHANGES	
-Documented in MTB		355	Document Specify One or More
-User/Operations-visible		BOS	MPM (Vol, Sect.)
Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		Salvager	PLMS (AN #)
-Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no		Ring Zero	MOSN (Sect.)
-Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same		Ring One	MPAM (Sect.)
<input type="checkbox"/> Worse		SysDaemon/Admin.	MSAM (Sect.)
-Replaces MCR		Runtime	
		X User Cmnd/Subr.	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason) X	

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

SUMMARY:

Fix bugs in the tape_mult I/O module.

REASONS:

Some of these bugs inhibit tape_mult from being used by system processes (backup, retriever) and from generating correct Multics system tapes.

DETAILED PROPOSAL:

1. Make "-system" and "-comment" attach description arguments work correctly.
2. Fix bug that, in certain cases, causes two (instead of one) EOF records to be written before the End of Reel record.
3. Fix bug that causes padded records to have the wrong format.
4. Fix bug that causes control request to get number of errors to fail.

TITLE: Fix bug in delete_		STATUS	DATE
AUTHOR: S. Herbst		Written	22 November 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 _____		Status	A 12/02/75
		Expires	06/06/76
		DOCUMENTATION CHANGES	
		Lib. Maint. Tools	
		Sys. Anal. Tools	
		Sys. Prog. Tools	
		355	
		BOS	
		Salvager	Document Specify One or More
		Ring Zero	MPM (Vol, Sect.)
		Ring One	PLMS (AN #)
		SysDaemon/Admin.	MOSN (Sect.)
		Runtime	MPAM (Sect.)
		User Cmmnd/Subr.	MSAM (Sect.)
Objections/Comments:			Info Segs
			Other (Name)
			None (Reason)documentation ok

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

SUMMARY:

Fix bug in delete_ that sometimes changes the value of the input argument switches (bit(6)).

Multics Change Request

TITLE: Fix bugs in walk_subtree command		STATUS	DATE	
AUTHOR: S. Herbst		Written	22 November 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	Category (Check One)	Status	A 12/02/75	
		Lib. Maint. Tools	Expires	06/02/76
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document Specify One or More	
	355	MPM (Vol, Sect.) Commands		
	BOS	PLMS (AN #)		
	Salvager	MOSN (Sect.)		
	Ring Zero	MPAM (Sect.)		
	Ring One	MSAM (Sect.)		
	SysDaemon/Admin.			
	Runtime			
	<input checked="" type="checkbox"/> User Cmmnd/Subr.			

Objections/Comments:

Notification in pending_changes is necessary.

Info Segs

Other (Name)

None (Reason) documentation ok

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

SUMMARY:

Fix two bugs in walk_subtree. These are the same fixes proposed in MCR 276 but not installed because set_acl did not work on MSF's (see #2):

1. change_wdir in the command line to be executed can cause the walk to continue in another directory. walk_subtree should be changed to restore the working directory to what it was before executing the command line, and then continuing the walk.
2. walk_subtree walks through MSF's and should be fixed not to.

IMPLICATION:

Incompatible change for the better.

TITLE: Fix -bf bug in dprint		STATUS	DATE
AUTHOR: S. Herbst		Written	24 November 75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> AIM <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)		
	<input type="checkbox"/> Lib. Maint. Tools	Status	A 12/02/75
	<input type="checkbox"/> Sys. Anal. Tools	Expires	06/02/76
	<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 #)		
<input type="checkbox"/> Ring Zero	MOSN (Sect.)		
<input type="checkbox"/> Ring One	MPAM (Sect.)		
<input type="checkbox"/> SysDaemon/Admin.	MSAM (Sect.)		
<input type="checkbox"/> Runtime	Info Segs		
<input checked="" type="checkbox"/> User Cmmnd/Subr.	Other (Name)		
	None (Reason) documentation ok		
Objections/Comments:			

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

SUMMARY:

Fix the dprint command to accept the -brief control argument at the end of the line.

REASON:

Brief mode applies to a message printed at the end, not to a particular pathname. Currently, the line:

```
dprint foo -bf
```

prints the error message:

```
dprint: Warning -- Control arguments following
last pathname are ignored
```

and -bf is not ignored.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1522
TITLE: Fix bug in cross_reference		STATUS: DATE
AUTHOR: Paul Green		Written: 11/20/75
		Status: A 12/02/75
		Expires: 05/20/76
Planned for System: MR 3.1		CATEGORY (check one)
Fixes Bug Number(s): CK97 ?		() Lib. Maint. Tools
Documented in MTB: not applicable		() Sys. Anal. Tools
Incompatible Change: no		() Sys. Prog. Tools
User/Operations-visible Interface Change: no		() 355
Coded in: (X) PL/I () ALM () other-see below		() BOS
Performance: () better (X) same () worse		() Salvager
<u>DOCUMENTATION CHANGES (specify one or more)</u>		() Ring Zero
MPM (vol,sect) MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		() SysDaemon/Admin
PLMs (AN#)		() Runtime
Info Segs		(X) User Command/Subr
Other		
<u>None (reason) no change</u>		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Fix bug in cross_reference which produced incorrect output when output segment exceeded 64K.

REASONS:

Bug should be fixed.

IMPLICATIONS:

We will be able to cross reference the entire library for the first time.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1523
TITLE: Log hardware errors in syserr log		STATUS DATE
AUTHOR: Larry Johnson		Written 11/21/75
		Status A 12/22/75
		Expires 05/21/76
Planned for System: MR 3.1		CATEGORY (check one)
Fixes Bug Number(s): not applicable		() Lib. Maint. Tools
Documented in MTB: not applicable		() Sys. Anal. Tools
Incompatible Change: no		() Sys. Prog. Tools
User/Operations-visible Interface Change: no		() 355
Coded in: (X)PL/I ()ALM ()other-see below		() BOS
Performance: ()better (X)same ()worse		() Salvager
DOCUMENTATION CHANGES (specify one or more)		(X) Ring Zero
MPM (vol,sect) MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		() SysDaemon/Admin
PLMs (AN#) yes		() Runtime
Info Segs		() User Command/Subr
Other		
OBJECTIONS/COMMENTS:		

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

Summary

Modify procedures which report hardware errors to use the syserr binary data mechanism to record error information.

Reasons

To simplify and standardize the logging of hardware errors for HEALS.

Detailed proposal

Change syserr calls in the following:

1. ioi_interrupt.pl1
2. parity_fault.pl1
3. ocdcm_.pl1
4. disk_control.pl1

Similar changes may be made to syserr calls in other modules that can report information of use to HEALS.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1524
TITLE: Add new features to mexp needed for FAST Fortran.		STATUS: DATE Written: 11/25/75
AUTHOR: Steve Webber		Status: A 12/02/75 Expires: 05/25/76
Planned for System: MR 3.1		CATEGORY (check one)
Fixes Bug Number(s): unreported		() Lib. Maint. Tools
Documented in MTB: not applicable		() Sys. Anal. Tools
Incompatible Change: no		() Sys. Prog. Tools
User/Operations-visible Interface Change: yes		() 355
Coded in: (X) PL/I () ALM () other-see below		() BOS
Performance: (X) 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 (attached)		() Runtime
Info Segs attached		(X) User Command/Subr
Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Add several new features to mexp needed for the FAST Fortran code generator interpreter. Also extend mexp to handle recursive expansion within conditionally expanded text.

REASONS:

Needed for FAST Fortran. Genenerally useful extensions.

IMPLICATIONS:

None.

DETAILED PROPOSAL:

The following pseudo-ops will be recognized:

- | | |
|-------------------------|---|
| <code>ifend_exit</code> | Can be used in place of <code>ifend</code> to stop conditional expansion of text. It instructs the macro expander to stop expanding text for the entire macro if the preceding conditionally expanded code was expanded. This allows for more flexibility in defining conditional macros. |
| <code>ifint</code> | conditionally expands the following text if the (first) argument to the pseudo-op is a decimal integer. A null argument is not |

treated as a decimal integer.

inint

conditionally expands the following text if
the (first) argument to the pseudo-op is not
a decimal integer or is null.

(6 lines follow; 27 lines in segment)

11/20/72 - mexp_changes

The mexp tool is being changed to accept three new pseudo-ops enabling more power in conditionally expanding text.

Rest of segment has 9 lines titled "New pseudo-ops". More help? yes

New pseudo-ops: The following pseudo-ops are being added

ifint to conditionally expand code if the (first) argument to the pseudo-op is a decimal integer.

inint to conditionally expand code if the (first) argument to the pseudo-op is not a decimal integer.

ifend_exit to stop expanding code for the entire macro. This pseudo-op is honored when it terminates the conditionally expanded text which is actually expanded (the condition is met).

- 8. dup causes the text up to the next dupend found in the text to be duplicated n times where n is the decimal value of the (first) parameter to the pseudo-operation.
- 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 corresponding to the argument position of the iteration argument for which the current iteration is being done (see "Examples" below).
- 11. &An is expanded to be the n+1'st argument to the mexp command.
- 12. ifarg if ifarg occurs in the context of an opcode or pseudo-operation it causes conditional expansion of the text up to the next ifend depending on whether or not the first parameter to the pseudo-operation is one of the arguments to the mexp command (other than the source name).

is &ln

length in char of operand

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

INSERT

The argument &0 expands to be the first label on the statement invoking a macro.

Any parentheses around a parameter are stripped off upon expansion. Parentheses used in this manner are treated as quoting characters.

Blanks cannot appear in a macro parameter list unless within a parenthesized parameter.

Iteration

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. (If no parameter number is specified, 1 is assumed.) Iterated arguments are scanned in the same manner as macro arguments and hence quoting can be done with the use of parentheses.

INSERT

14. `if end_exit` can be used in place of `ifend` to stop conditional expansion of text. It instructs the macro expander to stop expanding text of entire macro if the preceding conditionally expanded code was actually expanded.
15. `ifint` conditionally expands the following text if the (first) argument to the pseudo-op is a decimal integer. A null argument is not treated as a decimal integer.
16. `inint` conditionally expands the following text if the (first) argument to the pseudo-op is not a decimal integer or is null.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1525												
TITLE: Fix bugs introduced in the 27-2 command processor. AUTHOR: Steve Webber		<table border="1"> <tr> <th>STATUS</th> <th>DATE</th> </tr> <tr> <td>Written</td> <td>11/24/75</td> </tr> <tr> <td>Status</td> <td>A 12/02/75</td> </tr> <tr> <td>Expires</td> <td>05/24/76</td> </tr> </table>	STATUS	DATE	Written	11/24/75	Status	A 12/02/75	Expires	05/24/76				
STATUS	DATE													
Written	11/24/75													
Status	A 12/02/75													
Expires	05/24/76													
Planned for System: MR 3.1 Fixes Bug Number(s): unreported Documented in MTB: not applicable Incompatible Change: no User/Operations-visible Interface Change: no 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		<table border="1"> <tr> <th>CATEGORY (check one)</th> </tr> <tr> <td><input type="checkbox"/> Lib. Maint. Tools</td> </tr> <tr> <td><input type="checkbox"/> Sys. Anal. Tools</td> </tr> <tr> <td><input type="checkbox"/> Sys. Prog. Tools</td> </tr> <tr> <td><input type="checkbox"/> 355</td> </tr> <tr> <td><input type="checkbox"/> BOS</td> </tr> <tr> <td><input type="checkbox"/> Salvager</td> </tr> <tr> <td><input type="checkbox"/> Ring Zero</td> </tr> <tr> <td><input type="checkbox"/> Ring One</td> </tr> <tr> <td><input type="checkbox"/> SysDaemon/Admin</td> </tr> <tr> <td><input type="checkbox"/> Runtime</td> </tr> <tr> <td><input checked="" type="checkbox"/> User Command/Subr</td> </tr> </table>	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
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) MPAM (sect) MOSN (sect) MSAM (sect) PLMs (AN#) Info Segs Other None (reason) Bug fix only.														
OBJECTIONS/COMMENTS:														

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

SUMMARY:

Fix a bug in the 27-2 command processor which failed to parse lines correctly if they had trailing white space and did not end with a semicolon or newline character.

REASONS:

To get to the other side.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1526
TITLE: Install new temporary segment manager for use in Multics.		STATUS: DATE Written 11/20/75 Status <u>4/12/02/76</u> Expires 05/20/76
AUTHOR: Steve Webber		
Planned for System: MR 3.1		
Fixes Bug Number(s): not applicable		
Documented in MTB: not applicable		
Incompatible Change: no		
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 type="checkbox"/> same <input type="checkbox"/> worse		
<u>DOCUMENTATION CHANGES (specify one or more)</u>		<u>CATEGORY (check one)</u>
MPM (vol,sect) SWG	MPAM (sect)	<input type="checkbox"/> Lib. Maint. Tools
MOSN (sect)	MSAM (sect)	<input type="checkbox"/> Sys. Anal. Tools
PLMs (AN#)		<input type="checkbox"/> Sys. Prog. Tools
Info Segs attached	get, linkage, MPM sub-	<input type="checkbox"/> 355
Other	routine list in PLM 51.	<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
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Install a temporary segment manager to be used by standard commands which have need of temporary segments.

REASONS:

Useful program that can make better use of segments in the process directory by multiplexing their use among several programs.

IMPLICATIONS:

To take full advantage of the mechanism many commands will have to be changed. However, the commands would supposedly work faster and better. For example, edm and qedx could be used recursively as teco now can be.

DETAILED PROPOSAL:

There would be three new interfaces provided as follows:

get_temp_segments_ to return pointers to zero-length segments in the process directory to be used by the calling program as it sees fit.

release_temp_segments_ to return the temporary segments to the free pool of such. The command has the option of doing this, but it

defeats the purpose of the mechanism if it does not.

`list_temp_segments` to list the temporary segments currently in use as well as to give information about currently unused segments. The information listed would include (unique) name, segment number and, if being used, the name of the program using the segment.

See the attached SWG descriptions for the actual details of these interfaces.

list_temp_segments

list_temp_segments

Name: list_temp_segments

The list_temp_segments command lists the segments currently in the temporary segment pool managed by the get_temp_segments_ and release_temp_segments_ subroutines.

Usage

list_temp_segments -control_arg-

where control_arg may be the following:

-all, -a indicates that all temporary segments be listed. The normal mode is to list only those temporary segments currently assigned to some program.

Example

list_temp_segments -all

5 Segments, 2 Free

```
!BBBCdfgngffkkkl,temp.246 qedx
!BBBCdffddfdffkl,temp.247 qedx
!BBBCadffgffhnh,temp.253 (free)
!BBBCogdgfhfgfsf,temp.254 (free)
!BBBCvdvfgvdgvvv,temp.321 edm
```

r 1541 .163 12.433 13

get_temp_segments_

get_temp_segments_

Name: get_temp_segments_

The get_temp_segments_ subroutine is used ~~by commands~~ to acquire temporary segments in the process directory for whatever purpose the ~~command~~ *caller* may have. The segments returned are zero-length.

Usage

```
declare get_temp_segments_ entry (char (*), (*) ptr, fixed
    bin (35));
```

```
call get_temp_segments_ (command_name, ptrs, code);
```

where:

1. command_name is the name of the command requesting temporary segments. (Input)
2. ptrs is an array of returned pointers to the requested temporary segments. (Output)
3. code is a standard system status code. (Output)

Notes

The subroutine will create new temporary segments and add them to its pool of such if there currently are not enough available to satisfy the request. The temporary segments are created in the process directory with a unique name including the suffix ".~~etc~~". See the writeup for release_temp_segments_ for a description of how to return temporary segments to the free pool.

temp.### (where ### is an octal representation of the segments segment number)
The number of segments returned to the caller is determined by the bounds of the array "ptrs" above. (In PL/I terms, there will be $\text{ubound}(\text{ptrs}, 1) - \text{lbound}(\text{ptrs}, 1) + 1$ segments ~~allocated~~ returned.)

release_temp_segments_

release_temp_segments_

Name: release_temp_segments_

The release_temp_segments_ subroutine is used to return temporary segments acquired with the get_temp_segments_ subroutine to the free pool of such segments.

Usage

```
declare release_temp_segments_ entry (char (*), (*) ptr,  
fixed bin (35));
```

```
call release_temp_segments_ (command_name, ptrs, code);
```

where:

1. command_name is the name of the command releasing the temporary segments. (Input)
2. ptrs is an array of pointers to the temporary segments being released. (Input)
3. code is a standard system status code. (Output)

Note

The status code is returned non-zero if any of the segments being released were not assigned to the given command program. See the writeup for get_temp_segments_ for a description of how to acquire temporary segments.

Ver. 3
741022

MULTICS CHANGE REQUEST

MCR 1527

TITLE: Upgrade system tool hunt to search
archives and document in PLM.
AUTHOR: Steve Webber

STATUS: DATE:
Written: 11/25/75
Status: A/2/02/75
Expires: 05/25/76

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

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

DOCUMENTATION CHANGES (specify one or more)

MPM (vol,sect) MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#) AN51 (attached)
Info Segs
Other

OBJECTIONS/COMMENTS:

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

SUMMARY:

Upgrade the system tool hunt to search archives for the specified segment. Also, change the default directory used as the root of the tree to search from the root (>) to the user's current working directory.

REASONS:

Useful extensions.

IMPLICATIONS:

Any (system) user of hunt that assumes the root (>) as the default root of the tree to search will have to be changed.

DETAILED PROPOSAL:

See attached documentation.

hunt

hunt

Name: hunt

The hunt command searches a specified subtree of the hierarchy for all occurrences of a named segment, either free standing or included in an archive file. The segment(s) searched for can be specified by a star name. Any matching segments are reported.

Usage

hunt starname -root_of_tree- -control_args-

where:

1. starname is the (possibly star laden) name of the segment(s) to search for.
2. root_of_tree is the pathname of a directory to be interpreted as the root of the subtree in which to search for the specified segment(s). If no root_of_tree argument is specified, the hunt command searches the subtree rooted at the current working directory.
3. control_args may be chosen from the following list:
 - all, -a to report on finding links and directories as well as segments.
 - first to stop searching as soon as the first occurrence of the specified segment was found.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1528
TITLE: Redefine the format of gate segments in the system.		STATUS DATE
AUTHOR: Steve Webber		Written 11/25/75
		Status A 12/16/75
		Expires 05/25/76
Planned for System: MR 3.1		CATEGORY (check one)
Fixes Bug Number(s): not applicable		() Lib. Maint. Tools
Documented in MTB: not applicable		() Sys. Anal. Tools
Incompatible Change: no		() Sys. Prog. Tools
User/Operations-visible Interface Change: yes		() 355
Coded in: () PL/I (X) ALM (X) other-see below		() BOS
Performance: (X) 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#) AN 81 (to be written)		(X) Runtime
Info Segs		() User Command/Subr
Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Introduce a restriction on the format and design of system gate segments that are linked to by the standard system linkers. The change in the format is to restrict the entry point at location zero in the gate segment to be a standard entry which provides a mapping between entry point name and offset within the gate segment.

REASONS:

By providing this "actor" function, the user-ring linker, and the prelinker in particular, can search the definitions of inner ring gates. In addition, the ability to search the definitions is governed by the same access control mechanisms that govern whether or not the calling process can use the gate at all.

IMPLICATIONS:

When we do get around to moving the standard linker to the user ring, this problem will have been solved.

DETAILED PROPOSAL:

The proposal is to make the first location in each gate segment be an entry called as follows:

```
declare gate$0 (char (32) var, fixed bin (18));
```

```
call hcs_$initiate (<whatever>, "gate", "1", 1, 0, gatep, code);  
call cu_$ptr_call (gatep, <entrypoint name>, offset);
```

where:

offset will be returned as the value of the named entry point in the gate segment.

The same gate entry will also search for the name of an entry given an offset within the gate. This search is triggered if the first argument to the gate\$0 entry point is a zero length string.

Note that this mechanism obsoletes the procedure `get_defname`.

No change is needed to gate source segments to implement this change. The only thing which need be done is to regenerate the segments using new mexp macros.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1529
TITLE: Install a new standard system area management package.		STATUS: <u>Written</u>
AUTHOR: Steve Webber		DATE: <u>11/24/75</u>
Planned for System: MR 3.1		Status: <u>A 12/02/75</u>
Fixes Bug Number(s): not applicable		Expires: <u>05/24/76</u>
Documented in MTB: not applicable		CATEGORY (check one)
Incompatible Change: yes		() Lib. Maint. Tools
User/Operations-visible Interface Change: yes		() Sys. Anal. Tools
Coded in: () PL/I (<input checked="" type="checkbox"/>) ALM () other-see below		() Sys. Prog. Tools
Performance: (<input checked="" type="checkbox"/>) better () same () worse		() 355
DOCUMENTATION CHANGES (specify one or more)		() BOS
MPM (vol,sect)	MPAM (sect)	() Salvager
MOSN (sect)	MSAM (sect)	() Ring Zero
PLMs (AN#) 84		() Ring One
Info Segs yes		() SysDaemon/Admin
Other		(<input checked="" type="checkbox"/>) Runtime
OBJECTIONS/COMMENTS:		() User Command/Subr
Notification in pending changes necessary.		

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

SUMMARY:

Install a new area management package that changes the allocation and freeing algorithms to be more Multicious.

REASONS:

The current area management package uses the "buddy system" which is not a very good strategy for virtual memory systems. In particular, it requires initialization of regions of the area which are never used and hence page references that are unnecessary. In addition, it is difficult to extend an existent area, and impossible to have an area which is not a power of two words long. For these reasons, the standard area provided for system use (system_free_n_) must be made large enough initially for most practical situations. This causes excessive paging and the use of 64K AST entries, thereby causing undo thrashing on that size pool.

IMPLICATIONS:

Any programs that think they understand the format and algorithms of the current area structure, and depend on it, will no longer necessarily work. This class of programs is small and hopefully nonexistent.

Any permanent areas currently existing in the system will be supported. The current area management programs will be retained to manage such areas. Any new areas created by the system will be of the new format and hence use the new strategies. The header of the area will indicate whether it is a new or old style area.

DETAILED PROPOSAL:

The new area format is described in MTB 219 as updated by MTR yyy. satisfied fully within a single segment.

help -pn new_areas
(8 lines follow; 14 lines in segment)
11/25/75 - new_areas

The format of standard areas created by the system and used by PL/I for allocation is being changed. The new strategy no longer uses the buddy system and hence old-style areas will be incompatible with the new area management programs. The old-style areas will, however, continue to be supported.

Rest of segment has 6 lines titled "Area conversion". More help? yes
Area conversion: Any permanent areas that users have can be reformatted, in place, by the new tool convert_area. Although this is not immediately necessary, users are urged to reformat their areas as support for old-style (buddy system) areas will eventually be withdrawn. For a description of how to use convert_area type "help convert_area".

To: Distribution
From: Steve Webber
Subject: Results of New Area Design Review
Date: 11/6/75

Results of New Area Design Review

This memo gives the results of the design review held for the proposed new area format. The modifications and changes proposed at that meeting are incorporated in the current design which is described below where it differs from the original design.

Points Brought Out at the Design Review

1. It would be possible to restructure the (new) areas so that "headers" only, instead of "trailers" are used for each allocated block. This requires special casing the last allocated block (which must be special cased anyway) but makes unnecessary the touching of the last two words of a block until necessary.
2. It would be convenient to have two control bits in the area header (set via SWG interface) which are interpreted as "zero-block-on-allocation" and "zero-block-on-free".
3. It would be convenient to have a control bit in the area header (set via SWG interface) indicating that no blocks will be freed in the area. This allows a much faster allocation scheme which requires no storage per block to be used.
4. It would be convenient, for debugging purposes, to have a control bit in the area header (set via SWG interface) indicating that all free requests for the area be ignored. This prevents reuse of an allocated block.
5. It would be valuable if the area management code could be as safe as (reasonably) possible in light of possible asynchronous events such as QUITs. In particular, it should be designed so that critical code is isolated and protected from interruption.

6. It would be convenient if a command could be written to convert a current (buddy system) area into a new-style area. (Offsets, of course, must not change.)
7. It would be valuable to restrict multisegment areas to use by a single process. That is, guarantee consistency only in the case where a single process is manipulating them.
8. When the allocation program needs to start scanning another component of the area, should the component area pointer be determined by a pointer in the area header or as the result of an external call? The external call approach is more costly but also more general and allows more control over the management of area segments.
9. How should an area be located in the external world? Should a pointer to the area always point to the first component? Or should it be possible to treat a pointer to a component as a pointer to the entire area (as is possible)?
10. What are the actual tasks required to implement the new area scheme? What are the tasks required to take full advantage of it throughout the standard execution environment?
11. Should we use 1 free list or 17 as proposed?
12. It was pointed out that due to a restriction in the definition of the PL/I language that the proposed areas are not standard PL/I areas.

Changes Resulting from the Above Remarks

The new design (amended) incorporates nearly all of the proposed enhancements and simplifications. In particular the following are proposed:

1. "trailers" will be replaced by "headers".
2. The following control bits will be defined:
 - a. zero on allocation
 - b. zero on free
 - c. allocate assuming no freeing
 - d. don't free
 - e. extensible area
3. Inhibited code will be used where appropriate
4. A `convert_area` command and `convert_area_` subroutine will be provided for converting today's areas into new-style areas.

5. The next area pointer will be determined by external call.
6. A pointer to an area always points to the first component of the area.
7. There will be no "system" bit in the area header. The program "get_next_area_ptr_" will always be called to get a pointer to the next area component. It may need to create another component.
8. The program old_alloc_ (and friends) can go away immediately.
9. The lack of conformance to the PL/I standard for areas (which was universally laughed at) will be done anyway.
10. The new area management strategy will use 14 pools corresponding to sizes from 2^{*3} to 2^{*17} .

Tasks

The following list describes the task needed to be performed in order to implement and install the new area package and take advantage of it!

1. Write, debug, and meter the new area management routines.
2. Install the new routines on CISL (including renaming old routines, etc.)
3. Write, debug, and install an area conversion routine.
4. Write a complete set of info segments describing the various changes resulting from the new area management code.
5. Write the PLM documentation describing the internal workings of the area management code.
6. Design, code, debug, and document the SWG interfaces to the area management code.
7. Change the PL/I compiler to use operator calls for allocation and freeing.
8. Change pl1_operators_ to include the new operators needed by the compiler.
9. Convert the PL/I compiler to generate *system links for external variables.
10. Convert the FORTRAN compiler to generate *system links for common blocks.

11. Design, write, debug, and document the new PL/I external name manager (handler for *system links).
12. Convert the linker to use areas for combined linkage regions and to understand *system links.
13. Rewrite dump_ls and print_linkage_usage to understand new linkage region format. We probably also want a tool to dump everything in a combined linkage region even if it is not storage used by the linker.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1533
TITLE: Eliminate use of switches in salvager		STATUS DATE
AUTHOR: VanVleck		Written 12/08/75
		Status 12/02/75
		Expires 06/08/76
Planned for System: not applicable		
Fixes Bug Number(s): not applicable		CATEGORY (check one)
Documented in MTB: not applicable		() Lib. Maint. Tools
Incompatible Change: no		() Sys. Anal. Tools
User/Operations-visible Interface Change: no		() Sys. Prog. Tools
Coded in: (X) PL/I () ALM () other-see below		() 355
Performance: () better (X) 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#) salv		() Runtime
Info Segs		() User Command/Subr
Other		(X) salvager
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Change the salvager so that it gets its input from the operator command to BOS instead of from the processor switches.

The BOOT and SALV commands to BOS will be changed to take all typed arguments not processed by the command and put them on the INTK card in the configuration deck as 4-character ASCII fields. The salvager will obtain its arguments by looking at the configuration deck.

The parameters which the operator may specify are:

- LONG long salvager
- LOUD list names of directories
- NOPR no printer available, don't print

Default for all these options is OFF.

REASONS:

The current method of salvager control is awkward and does not lend itself to conditional testing and execution, such as may be needed in RUNCOM files

for unattended system operation.

IMPLICATIONS:

In order to minimize operational impact the switches will still be read for a compatibility period and OR'ed into the options set by the command line.

The ability to change the mode of salvager operation into and out of LONG and LOUD modes will be lost. It is felt that this undocumented "feature" is no great loss.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1534
TITLE: BOS changes for unattended operation	STATUS: Written	DATE: 12/08/75
AUTHOR: VanVleck	Status: A	12/02/75
	Expires:	06/08/76
Planned for System: not applicable	CATEGORY (check one)	
Fixes Bug Number(s): not applicable	<input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input checked="" 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 Command/Subr	
Documented in MTB: MTB-152		
Incompatible Changes: no		
User/Operations-visible Interface Change: no		
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		
DOCUMENTATION CHANGES (specify one or more)		
MPM (vol,sect)	MPAM (sect)	
MOSN (sect)	MSAM (sect)	
PLMs (AN#)	BOS	
Info Segs		
Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Modify BOS toehold program to contain one word of 36 one-bit flags, accessible to both BOS and Multics.

Modify several BOS programs to maintain these flags.

The following flags will be provided:

<u>BOS name</u>	<u>Multics name</u>
AUTO	auto_reboot
CRASH	system_crashed
BOOTOK	boot_successful
MANRTB	manual_rtb

The IF command will be able to test these flags and cause conditional execution of RUNCOM files.

REASONS:

Communication between BOS and Multics must be expanded in order to support automatic recovery mode. In particular, the flags which tell the system what to do next must be writeable by BOS as well as by Multics, so that the operator can always disable this mode or re-enable it.

IMPLICATIONS:

Unattended operation of the system (now practiced by several installations) need not cease every time the system crashes.

DETAILED PROPOSAL:

1. Add flag word to BOS toehold at fixed location.
2. Modify BOOT command to
 - a) Set BOOTOK to OFF
 - b) Set CRASH to ONJust before giving control to the system tape.
3. Modify toehold to set MANRTB to ON for operator action.
4. Modify IF command to test flags.
5. Write new command SET to turn flags on and off.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1537
TITLE: New Implementation of tty_write		STATUS DATE
AUTHOR: Robert S. Coren		Written 12/08/75
Planned for System: MR 3.1		Status A12/02/75
Fixes Bug Number(s): not applicable		Expires 06/08/76
Documented in MTB: 234		CATEGORY (check one)
Incompatible Change: no		() Lib. Maint. Tools
User/Operations-visible Interface Change: yes		() Sys. Anal. Tools
Coded in: <input checked="" type="checkbox"/> PL/I () ALM <input checked="" type="checkbox"/> other-see below		() Sys. Prog. Tools
Performance: <input checked="" type="checkbox"/> better () same () worse		() 355
DOCUMENTATION CHANGES (specify one or more)		() BOS
MPM (vol,sect) sub,tty_ MPAM (sect)		() Salvager
MOSN (sect) MSAM (sect)		<input checked="" type="checkbox"/> Ring Zero
PLMs (AN#)		() Ring One
Info Segs tty_changes.info		() SysDaemon/Admin
Other		() Runtime
OBJECTIONS/COMMENTS:		() User Command/Subr

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

SUMMARY: Implement new design of tty_write described in MTB 234.

REASON: New design is faster and makes eventual user-substitutable tables possible.

IMPLICATIONS: 1. Warning message ("EOP") will be printed when screen or page is full (see info segment)

2. The status code error_table\$action_not_performed will be returned if the "printer_on" or "printer_off" control operation is requested for a terminal that lacks printer control (as described in another MCR). Installation of this change to the TTY DIM therefore depends on an Answering Service which checks for the code.

DETAILED PROPOSAL: Implement design described in MTB 234. In addition, rename the module containing the various conversion/translation tables from tty_ctl to tty_tables.

Note: tty_tables coded in mexp; minor 355 changes coded in 355map.

tty_changes.info

12/01/75

A new version of MCS has been installed which incorporates the following changes:

- 1) The status code `error_table$action_not_performed` is returned if the "printer_on" or "printer_off" control operation is requested for a terminal which does not have the printer_on/off feature. Formerly, a status code of zero was returned.
- 2) If the terminal's page length is non-zero, then when the line count reaches the specified maximum, the warning string "EOP" will be printed, and output will be suspended until a form-feed character is input. Formerly, output was suspended but no warning was printed.

Excerpt from MPM description of tty - control operation

`read_status` tells whether or not there is any type-ahead input waiting for a process to read. The `info_ptr` should point to the following structure that is filled in by the call:

```
dcl 1 info_structure      aligned,  
    2 ev_chan            fixed bin(71),  
    2 input_available    bit(1);
```

where:

1. `ev_chan` is the event channel used to signal the arrival of input.
2. `input_available` indicates whether input is available.
"0"b no input
"1"b input

`quit_enable` causes quit signal processing to be enabled for this device. (Quit signal processing is initially disabled.)

`quit_disable` causes quit signal processing to be disabled for this device.

`start` causes a wakeup to be signalled on the event channel associated with this device. This request is used to restart processing on a device whose wakeup may have been lost or discarded.

`printer_off` causes the printer mechanism of the terminal to be temporarily disabled if it is physically possible for the terminal to do so; if it is not, the status code `error_table$action_not_performed` is returned.

`printer_on` causes the printer mechanism of the terminal to be reenabled.

`wru` initiates the transmission of the device's answerback, if it is so equipped. This operation is allowed only for the process that originally attached the device (generally the initializer process). The answerback may subsequently be read by means of the `get_chars` input/output operation.

`store_id` stores the answerback identifier of the terminal for later use by the process. The `info_ptr` should point to a `char(4)` variable,

Excerpt from MPM description of tty - modes operation

- crecho.)
- tabecho, ^tabecho specifies that the appropriate number of spaces are to be echoed when a horizontal tab is typed. (Default is off; the same restriction applies as for crecho.)
- echoplex, ^echoplex specifies that all characters typed on the terminal are to be echoed. (Default is off; the same restriction applies for crecho.)
- fulldpx, ^fulldpx specifies that the terminal is to be allowed to receive and transmit simultaneously. (Default is off; this mode is automatically turned on and off when echoplex is turned on and off.)
- capo, ^capo specifies that all lowercase letters are to be output in uppercase. If edited mode is on, uppercase letters are printed normally; if edited mode is off and capo mode is on, uppercase letters are preceded by an escape (\) character.
- llg specifies the length in character positions of a terminal line. If an attempt is made to output a line longer than this length, the excess characters are placed on the next line. (Default line length is 130 for devices similar to IBM 1050s, 125 for IBM 2741s, 88 for Teletype Model 37, 118 for GE TermiNet 300s, 80 for ARDS, 72 for Teletype Models 33 and 35, 132 for Teletype Model 33, and 79 for ASCII devices.)
- plg specifies the length in lines of a page. When an attempt is made to exceed this length, a warning message is printed. When the user types a form-feed character, the output continues with the next page. If the page length is zero, end-of-page checking is disabled. (Default page length is 50 for ARDS-like terminals, and zero for all other terminals.)

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1538
TITLE: Add create_data_segment tool to system.		STATUS: <u>Written</u> DATE: <u>11/25/75</u>
AUTHOR: Steve Webber		Status: <u>A12/02/75</u>
Planned for System: not applicable		Expires: <u>05/25/76</u>
Fixes Bug Number(s): not applicable		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 Command/Subr
Documented in MTB: not applicable		
Incompatible Change: no		
User/Operations-visible Interface Change: no		
Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other-see below		
Performance: <input type="checkbox"/> better <input type="checkbox"/> same <input type="checkbox"/> worse		
DOCUMENTATION CHANGES (specify one or more)		
MPM (vol,sect) MPAM (sect)		
MOSN (sect) MSAM (sect)		
PLMs (AN#) AN51 (attached)		
Info Segs		
Other		
OBJECTIONS/COMMENTS:		
This will be upgraded to SSS after some experience with it has been gained.		

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

SUMMARY:

Add the tool create_data_segment (cds, even though it is now a tool) to the system. This command is an interface to cds-type programs for creating data segments with use of the create_data_segment_subroutine. The command works as a normal translator looking for source of the form "foo.cds" and creating an output segment named "foo". The command calls upon the PL/I compiler to compile the given source and then executes it to have the data segment created.

REASONS:

A very useful tool for creating data object segments which has more convenience, structure, and beauty than the alternative, ALM.

IMPLICATIONS:

The library installation tools should be upgraded to handle source segments with a suffix of ".cds".

DETAILED PROPOSAL:

See attached writeup.

create_data_segment

create_data_segment

Names: create_data_segment, cds

The create_data_segment command translates a create_data_segment source program (CDS program) into an object segment. A listing segment is optionally created. These results are placed in the user's working directory. This command cannot be called recursively.

The source for create_data_segment programs is standard PL/I with the restriction that the last executable statement be a call to the subroutine create_data_segment_. The create_data_segment_ program is described elsewhere in this manual and basically creates a standard object segment from PL/I data structures passed to it as parameters. These data structures can be initialized with arbitrarily complex PL/I statements in the CDS program.

Usage

create_data_segment path -control_arg-

where:

1. path is the pathname of a CDS segment that is to be translated into an object segment. If path does not have a suffix of cds, then one is assumed. However, the suffix cds must be the last component of the name of the source segment.
2. control_arg can be the following:
 - list, -ls produces a source listing of the CDS program used to generate the data segment followed by object segment information (as printed by the print_link_info command) about the actual object segment created.

Notes

Since the create_data_segment command invokes the PL/I compiler to first compile the CDS segment, any errors that the compiler finds are reported by its standard technique. If any errors with a severity greater than 2 occur, the CDS run is aborted and no object segment is created.

TITLE: Fix more bugs in Answering Service		STATUS	DATE
AUTHOR: Paul Green		Written	11/25/75
-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>3.1</u> -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 18/02/75
		Expires	06/02/76
		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 type="checkbox"/> Ring One <input checked="" 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 #)	ANSI
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	MOH; AM81
		None (Reason)	

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

SUMMARY:

1. Fix MAXUNITS operator command to reject zero as an invalid value,
2. Fix a bug in lg_ctl that prevents a user from being logged-in interactively, and as a daemon, at the same time,
3. Fix the -terminal_type login argument to work over the Network,
4. Fix absentee not to get confused when a user deletes a request while the absentee process is logged-in,
5. Install a tool to dump an absentee data base,
6. Make the "Try again at 0900" message include the day name if it is not the same as the current day.

REASONS:

1. Confuses the system terribly. The operator should have typed WORD XXX to shut off further logins.
2. Bug.
3. Bug.
4. Bug.
5. Will help the next time we have an absentee problem.
6. System shouldn't say "Try again at 0900" if 0900 is two days from now...

IMPLICATIONS:

1. Minor operator interface change. MOSN will be published.
- 2-6. Compatible changes.

DETAILED PROPOSAL:

- 1-4. Change the relevant answering service program.
5. Program is named "dump_abs_data" and takes no arguments. Will be installed in tools and documented in System Tools PLM (and/or Answering Service PLM).
6. Same as 1-4, above.

dump_abs_data

dump_abs_data

Name: dump_abs_data

This command prints the contents of the segment >system_control_1> absentee_data on the user_output switch. The ready and skip_list for each queue is printed, and the free list and defer list (which apply to all queues) is printed. Then each entry is printed. This command is intended as an aid to debugging the absentee facility, and is not of general use.

Usage

dump_abs_data

DRAFT: SUBJECT TO CHANGE

AN51 - System Tools

maxunits, maxu

This command alters the maximum number of load units accomodated by the system. If it is set to below the current number of units, no users are bumped, but only those users with guaranteed login privileges can log in. Type:

maxu NNN

to set the maximum number of load units to NNN/10. NNN must be greater than zero.

To cause the system to set ...

DRAFT: SUBJECT TO CHANGE

6-36

AM81 - MOH

Multics Change Request

TITLE: Install pll_operators_ to Interface with New Area Package		STATUS	DATE
AUTHOR: R. A. Barnes		Written	25 November 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 3.1 -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 12/02/75
	<input type="checkbox"/> Lib. Maint. Tools	Expires	06/02/76
	<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.)	
<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 checked="" type="checkbox"/> Runtime	Other (Name)		
<input type="checkbox"/> User Cmmnd/Subr.	None (Reason) no visible change		
Objections/Comments:			

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

SUMMARY:

Make changes necessary for installation of new area package and to prepare for the generation of operator calls by the compiler for allocate, free, and empty.

IMPLICATIONS:

None.

- DETAILED PROPOSAL:
1. Add segdef for call_signal_ used by area package.
 2. Add 4 new entries to the transfer vector which will be used by the MR4.0 PL/I compiler for allocate and free statements and the empty builtin function.

Multics Change Request

TITLE: Fix bug in fs_get		STATUS	DATE	
AUTHOR: E. Stone		Written	25 November 75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> AIM <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 <u>1478</u>	Category (Check One)		Status <u>A 12/02/75</u>	
		Lib. Maint. Tools	Expires <u>06/02/76</u>	
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
	355	MPM (Vol, Sect.)		
	BOS	PLMS (AN #)		
	Salvager	MOSN (Sect.)		
	<input checked="" type="checkbox"/> Ring Zero	MPAM (Sect.)		
	Ring One	MSAM (Sect.)		
	SysDaemon/Admin.			
	Runtime			
	User Cmnd/Subr.			

Objections/Comments:	Info Segs
	Other (Name)
	None (Reason) <input checked="" type="checkbox"/>

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

SUMMARY:

In system 27.1 a new version of fs_get was installed. Formerly the entry point fs_get\$brackets returned the extended ring brackets when called on behalf of a directory segment. The 27.1 version omitted filling in the ring brackets argument for directories.

PROPOSAL:

Change the present version of fs_get to return the extended ring brackets for directories.

Multics Change Request

TITLE: Retriever operational change		STATUS	DATE	
AUTHOR: A. Kobziar		Written	11-26-75	
-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	
		Lib. Maint. Tools	Expires	
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355	MPM (Vol, Sect.)	
		BOS	PLMS (AN #)	
		Salvager	MOSN (Sect.)	
		Ring Zero	MPAM (Sect.)	
		x Ring One	MSAM (Sect.)	
		SysDaemon/Admin.	Info Segs	
	Runtime	Other (Name)		
	User Cmnd/Subr.	None (Reason)		
Objections/Comments:				

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

Summary: Change retrieval of directories to attempt to move the amount of quota on the tape record to the directory rather than ignore the quota information entirely when the -quota option is not specified.

Reason: Retrievals (without -quota) of upgraded directories leave these directories security out-of-service since their quota is zero. Retrievals with -quota allow users to obtain unauthorized quota. A retrieval which attempts to move quota solves this problem. Also if the retrieval implicitly created a (missing) parent then the move_quota would fail, while a retrieval with -quota would restore quota on a directory whose parent had zero quota, creating an inconsistency in the hierarchy.

TITLE: IO Daemon Accounting Fixes		STATUS	DATE	
AUTHOR: Mary A. Braida		Written	11-25-75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.1 -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	
	Lib. Maint. Tools		Expires	
	Sys. Anal. Tools		A 12/02/75	
	Sys. Prog. Tools		06/02/76	
	355		DOCUMENTATION CHANGES	
BOS		Document	Specify One or More	
Salvager		MPM (Vol, Sect.)		
Ring Zero		PLMS (AN #) IO Daemon (unwritten)		
Ring One		MOSN (Sect.)		
<input checked="" type="checkbox"/> SysDaemon/Admin.		MPAM (Sect.)		
Runtime		MSAM (Sect.)		
User Cmnd/Subr.		Info Segs		
Objections/Comments:		Other (Name)		
		None (Reason)		

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

Summary: Change the IO Daemon to be more accurate in the accounting for restarted requests. Thus, if a request has been restarted after m of n copies have been completed, charges for n-m-1 copies will be computed after the restart.

Reasons: This change will cause the user to be charged for all his copies since presently the user would not be charged for n-m-1 copies after the restart.

TITLE: change object_info_		STATUS	DATE
AUTHOR: E. Stone		Written	26 November 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 12/02/75
		Expires	06/02/76
		DOCUMENTATION CHANGES	
Category (Check One)		Document	Specify One or More
<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.		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	X

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

SUMMARY:

Change object_info_ to include the length of the definition section as part of the length of the linkage section for those object segments whose definitions are in the linkage section.

REASONS:

This proposed change came as a result of making modifications to the generate_mst command. A side effect of making this command conform to system programming standards (changing it to call object_info_ rather than decode_object_) was that generate_mst had to special case alm object segments containing movedef pseudo-ops. Since object_info_ already special cases this situation, I consider it better to put this knowledge in object_info_ rather than the program generating mst's.

IMPLICATIONS:

A print_link_info for these segments will be different. There are six such segments currently installed--all of which are on the system tape. The feature of placing the definitions in the linkage section is not a frequently used one and should have no effect on the average user.

TITLE: vfile_changes for new file type		STATUS	DATE
AUTHOR: M. Asherman		Written	24 November 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 231 -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	12/2/75
		Expires	06/02/76
		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.)	I/O, iox, vfile
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason)	

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

SUMMARY:

Make changes in vfile_ to implement new file type, operations, and attach options.

REASONS:

Required for Fortran/Basic I/O in FAST.

IMPLICATIONS:

Centralized, interchangeable support of Fortran/Basic files.

Simple direct-access to non-indexed files.

DETAILED PROPOSAL:

Outlined in MTB 231, with modifications noted in MTR 107, and further amended as follows:

1. "-must_exist" option renamed "-old".
2. restriction to single segment files is specified by "-ssf" option.
3. calls to msf_manager_ are avoided except when file is an msf and -ssf option not specified.

Multics Change Request

TITLE: Test for non-interactive use of command_query_		STATUS	DATE	
AUTHOR: S. Herbst		Written	10/21/75	
-Coded in: <input checked="" type="checkbox"/> XPL/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 _____	Category (Check One)	Status	P 11/24/75 A 12/19/75	
		Lib. Maint. Tools	Expires	06/09/76
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
		355	MPM (Vol, Sect.)	
	BOS	PLMS (AN #)		
	Salvager	MOSN (Sect.)		
	Ring Zero	MPAM (Sect.)		
	Ring One	MSAM (Sect.)		
	SysDaemon/Admin.	Info Segs		
	Runtime	Other (Name)		
	<input checked="" type="checkbox"/> User Cmmd/Subr.	None (Reason)	doc ok	
Objections/Comments:				

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

SUMMARY:

Change command_query_ to signal command_query_error (the condition signalled by command_query_ today if a yes or no answer was requested but some other answer was read in) instead of asking the question if iox_\$ resetread returns the error code error_table_\$no_operation.

REASONS:

If the resetread could not be performed (a non-interactive device or I/O module such as abs_io_ is being used), the answer read in is probably not correct.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1532
TITLE: Use vfile_ for certain answering service files.		STATUS DATE
AUTHOR: VanVleck		Written 11/23/75
		Status 12/2/75 12/9
		Expires 05/23/76
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		() Sys. Prog. Tools
User/Operations-visible Interface Change: no		() 355
Coded in: (X) PL/I () ALM () other-see below		() BOS
Performance: () better (X) 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#) AN66		() Runtime
Info Segs		() User Command/Subr
Other		(X) admin
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Convert the pnt and the user_registration_file to PL/I direct keyed files.

REASONS:

The pnt and the user_registration_file data bases are used by the system administration software to register users. These data bases tend to grow steadily as users are registered; access control considerations cause them to shrink rarely, if ever. Each of these data bases is currently implemented by a single large segment and an associated hash table segment. We are now approaching the limits of the single-segment implementation, and it seems better to use existing software than to extend the current scheme.

The current hash program is a simple-minded adaptation of the hash program used for directories embedded within a hash segment manager program. vfile_, on the other hand, is a central and important part of the system runtime; it has salvaging and locking and all sorts of fancy features.

IMPLICATIONS:

If the hash tables are counted, the amount of space occupied is about equal.

Access time is about twice as slow. To look up 3800 entries via the current pnt hash table takes about 1.9 cpu seconds; with keyed direct input it takes about 3.8. The comparison is even less favorable to vfile if we

notice that once a record is located via the current structure it can be modified in place, while with vfile a rewrite operation must be done. However, 2 seconds of CPU or even 4 for every 3600 logins is negligible.

If another bug causing randomization of keys showed up in vfile when the answering service was depending on it to manage the password file, it would be a major catastrophe.

DETAILED PROPOSAL:

Most of the changes are straightforward. The modules lg_ctl_, new_user, pass_util, list_extra_personids, and print_pnt are affected. The program hash_ can be eliminated, as well as the ring 4 copy of hash_index and the test program hash_table. Minor changes may be made to several other programs.

The programs which install new versions of the PNT, up_pnt_ and up_sysctl_, currently depend on the ability to copy the new contents of a segment into the already-initiated copy of a segment with one statement. We can't do this for MSP's, so a different mechanism will have to be coded. Copying the whole PNT is probably a waste of time anyway, though, since there are no global changes to the file and since entries are (almost) never removed. New_user, install, up_sysctl_, and up_pnt_ will be changed to support an incremental installation operation which just installs a single PNT entry, or set of entries.

The current hash tables have aliases as well as personids as keys. The multiple keys feature to be added to vfile_ for MR4.0 will be used for aliases.

Modes which allowed vfile_ to furnish a direct pointer to the record rather than requiring it to be copied and rewritten would improve the efficiency of the answering service.

Ver. 3 741022	MULTICS CHANGE REQUEST	MCR 1535
TITLE: Automatic crash recovery mode		STATUS DATE
AUTHOR: VanVleck		Written 11/23/75
		Status PA 12/9
		Expires 05/23/76
Planned for System: not applicable		CATEGORY (check one)
Fixes Bug Number(s): not applicable		
Documented in MTB: MTB-152		
Incompatible Change: no		
User/Operations-visible Interface Change: no		
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		
DOCUMENTATION CHANGES (specify one or more)		
MPM (vol,sect)	MPAM (sect)	
MOSN (sect)	MSAM (sect)	
PLMs (AN#) hardcore		
Info Segs		
Other		
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Modify system initialization programs to respect and maintain the flags in the BOS toehold which tell if the system is in automatic crash recovery mode. See the preceding MCR for an explanation of the flags.

If the system makes it up to answering service initialization, the boot_successful flag is set ON. If BOS regains control and finds boot_successful OFF, the runcom knows that the system crashed during bootload, and so can turn off auto_reboot mode to prevent loops. A heuristic in answering service initialization will set auto_reboot OFF if the system seems to be crashing too frequently.

REASONS:

Multics initialization should respect the unattended mode switches.

IMPLICATIONS:

Various special commands can be written to read and set the toehold flags. These commands can be invoked by the system_start_up.ec.

DETAILED PROPOSAL:

1. Modify system header to make toehold writeable in ring 0.

2. Add new gate entries
 hphcs_\$set_bos_flags
 phcs_\$get_bos_flags
3. Modify `system_startup_` to accept a parameter from the `BOOT` command line. If this parameter is "STAR" then do not read a line from the terminal in ring 1, but continue with startup.
4. Provide code in answering service initialization (as `meter $init`) to set `flags.auto_reboot` OFF if there are (say) 5 crashes in 30 minutes without operator intervention. The numbers will be installation parameters.
5. Change answering service initialization to set `flags.boot_successful` ON when the system is successfully started.
6. Change answering service shutdown to set `flags.system_crashed` OFF before returning to BOS.
7. Change `system_control_` and `system_startup_` to set `flags.system_crashed` OFF if returning to BOS manually.

Ver. 3	MULTICS CHANGE REQUEST	MCR 1536
741022		
TITLE: New subroutine error reporting mechanism	STATUS	DATE
AUTHOR: VanVleck	Written	11/25/75
	Status	12/21/76
	Expires	05/25/76
Planned for System: not applicable	CATEGORY (check one)	
Fixes Bug Number(s): not applicable	<input type="checkbox"/> Lib. Maint. Tools	
Documented in MTB: MTB-215	<input type="checkbox"/> Sys. Anal. Tools	
Incompatible Change: yes	<input type="checkbox"/> Sys. Prog. Tools	
User/Operations-visible Interface Change: no	<input type="checkbox"/> 355	
Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other-see below	<input type="checkbox"/> 80S	
Performance: <input type="checkbox"/> better <input checked="" type="checkbox"/> same <input type="checkbox"/> worse	<input type="checkbox"/> Salvager	
DOCUMENTATION CHANGES (specify one or more)		
MPM (vol,sect) SWG	MPAM (sect)	<input type="checkbox"/> Ring Zero
MOSN (sect)	MSAM (sect)	<input type="checkbox"/> Ring One
PLMs (AN#)		<input type="checkbox"/> SysDaemon/Admin
Info Segs		<input type="checkbox"/> Runtime
Other		<input checked="" type="checkbox"/> User Command/Subr
OBJECTIONS/COMMENTS:		

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

SUMMARY:

Install sub_err_ as described in MTB-215.

Change com_err_ to say "Code 1" instead of "Code 1 not found in error_table_."

Modify interpret_info_struct_ slightly to produce a better standard message for sub_error_ conditions handled by the standard default handler.

REASONS:

This new feature will allow subsystems and subroutines to report errors without explicitly using the terminal and without forcing them to compress all error information into a single error code.

IMPLICATIONS:

User default handlers which catch a sub_error_ condition must insure that the error message is printed.

sub_err_.info

(date) sub_err_

Function: sub_err_ is used to report subsystem and subroutine errors. It signals the condition "sub_error_".

Syntax: dcl sub_err_ entry options (variable);
call sub_err_ (code, name, flag, infop, retval, ctl_string, args);

Arguments:

code status code for error
name char (*) module name
flag "h" halt, "c" continue, "s" stop (no return)
infop ptr to further info, if any
retval returned 0 by standard environment
ctl_string loa_ format control string for message
args any arguments for output message

Message: The output message will look like this:
name error by callernamellocation
Status code message. Message from ctl_string.

Handlers: All any_other handlers should be able to handle the "sub_error_" signal. They must insure the message is printed and return or abort according to the switches.

Info Structure: the software_info_structure which handlers may obtain from find_condition_info_ has the following format:

```
dcl 1 info aligned based (software_info_ptr),
    2 length fixed bin,
    2 version fixed bin init (2),
    2 action_flags aligned,
    3 cant_restart bit (1) unal,
    3 default_restart bit (1) unal,
    3 pad bit (34) unal,
    2 info_string char (256) var,
    2 code fixed bin (35),
    2 retval fixed bin (35),
    2 name char (32),
    2 infop ptr;
```

Name: sub_err_

This program is called by subroutines which wish to report an unexpected situation, without usurping the calling environment's responsibility for the content of and disposition of the error message, and the choice of what to do next. The caller specifies an identifying message and may specify a status code. Switches which describe whether and how to continue execution and a pointer to further information may also be passed to sub_err_. The environment which invoked the subroutine caller of sub_err_ may intercept and modify the standard system action taken when sub_err_ is called.

Usage:

```
dcl sub_err_ entry options (variable);
```

```
call sub_err_ (code, name, flags, infop, retval,  
              ctl_string, loa_args);
```

where

- 1) code is a status code describing the reason for calling sub_err_. code should be declared fixed bin (35). (Input)
- 2) name is the name of the subsystem or module on whose behalf sub_err_ is called. name should be declared as a nonvarying character string. (Input)
- 3) flags describe how and whether restart may be attempted. Flags should be declared as a nonvarying character string. (Input)

The following values are permitted:

"h"	halt at command level after printing message. Resume if start is typed.
"c"	continue after printing message.
"s"	stop. Attempt to restart will raise the illegal_return condition.
- 4) infop is an optional pointer to information specific to the situation. The standard system environment does not use this pointer, but it is provided for the convenience of other environments. infop should be an aligned pointer. (Input)
- 5) retval is a return value from the environment to which the error was reported. The standard system environment sets this value to zero. Other environments may set retval to other values, which may be used to select recovery strategies. retval should be declared fixed bin (35). (Input/Output)
- 6) ctl_string is an loa_ format control string which defines the

message associated with the call to sub_err_. Consult the description of loa_ in AG93. cti_string should be declared as a nonvarying character string. (Input)

7) loa_args are any arguments required for conversion by cti_string. (Input)

Operation

Sub_err_ proceeds as follows: the structure described below is filled in from the arguments to sub_err_, and the system subroutine signal_ is called to raise the "sub_error_" condition.

When the standard system environment receives a sub_error_ signal, it prints a message of the format

```
name error by subrname/location
Status code message. Message from cti_string.
```

The standard environment then sets retval to zero and returns, if "c" was specified; otherwise it calls the listener. If "start" is typed, the standard environment will return to sub_err_, which will return to the subroutine caller of sub_err_ unless "s" was specified. If "s" was specified, sub_err_ will signal "illegal_return."

Handler operation

All handlers for the "any_other" condition must either pass the "sub_error_" condition on to another handler, or else must handle the condition correctly. Correct handling consists of printing the error message and of respecting the "cant_restart" and "default_restart" flags, unless the environment deliberately countermands these actions (for example, for debugging purposes).

If an application program wishes to call a subsystem which may report errors by sub_err_, and wishes to replace the standard system action for some classes of sub_err_ calls, the application should establish a handler for the "sub_error_" condition by a PL/I ON-statement. When the handler is activated as a result of a call to sub_err_ by some dynamic descendant, the handler should call find_condition_info_ to obtain the "software_info_ptr" which will point to a structure with the following declaration.

```
dcl 1 info aligned based (software_info_ptr),
    2 length fixed bin,
    2 version fixed bin,
    2 action_flags aligned,
    3 cant_restart bit (1) unal,
    3 default_restart bit (1) unal,
    3 pad bit (34) unal,
    2 info_string char (256) var,
```

```
2 code fixed bin (35),
2 retval fixed bin (35),
2 name char (32),
2 infop ptr;
```

where

length	is the size of the structure in words.
version	is the version number of the structure. This is version 2.
cant_restart	is "1"b if the condition cannot be restarted.
default_restart	is "1"b if the standard environment will print the message and continue execution without calling the listener.
pad	is padding
info_string	is the converted message from cti_string and loa_args.
code	is the status code.
retval	is the return value. The standard environment sets this value to zero.
name	is the name of the module encountering the condition.
infop	is a pointer to additional information associated with the condition.

The handler should check info.name and info.code to make sure that this particular call to sub_err_ is the one desired, and if not call continue_to_signal_. If the handler determines that it wishes to intercept this call to sub_err_, the info structure will provide the message as converted, switches, etc. Any change made to the value of info.retval will be returned to the caller of sub_err_ if control returns to sub_err_.

When to Use

Multics conventions currently forbid subroutines which may be called by many different programs from performing output unless that is their primary purpose. The reason for this rule is the "principle of transparency," which requires that the subroutine be usable in environments which do not have standard I/O attachments, and in environments which wish to use the subroutine without obtaining any output. In particular, subroutines are currently forbidden to use com_err_ to report status. The standard method for reporting status is to supply an additional argument to the subroutine which will be set to zero or to a standard status code by the subroutine.

The caller of such a subroutine must have some knowledge of the cases in which status codes are returned. Often, the calling program has the choice of including a series of tests for each of the possible statuses recognized by the subroutine, or of simply assuming that any nonzero status code indicates that the routine failed. When a status code is returned, the calling program often wishes to produce a message describing the situation. But in some cases, the subroutine can recognize so many different situations that the calling program will be unable to produce a helpful message without additional communication between the calling program and the subroutine.

Subroutines which can detect multiple errors (such as compilers) have an even more difficult problem. The returning of a status code is suited only to the detection of single errors. Requiring the calling program to allocate storage for a usually null array of status indicators or status messages seems uneconomical; and saving the messages in storage allocated by the subroutine encounters other problems if multiple invocations of the routine may exist in the same process.

General-purpose subsystems or subroutines which can be called in a variety of I/O and error handling environments should report the errors which they detect by calling `sub_err_`. The caller of `sub_err_` can specify that the normal action to be taken is to continue ("c"), halt at command level ("h"), or stop ("s").

TITLE: vfile_bug fixes		STATUS	DATE	
AUTHOR: M. Asherman		Written	24 November 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) <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	
		Lib. Maint. Tools	Expires	
		Sys. Anal. Tools	P 12/2/75 A 12/9/75 06/9/76	
		Sys. Prog. Tools	DOCUMENTATION CHANGES	
		355	Document	Specify One or More
		BOS	MPM (Vol, Sect.)	
		Salvager	PLMS (AN #)	
		Ring Zero	MOSN (Sect.)	
		Ring One	MPAM (Sect.)	
		SysDaemon/Admin.	MSAM (Sect.)	
	Runtime			
	<input checked="" type="checkbox"/> User Cmnd/Subr.			
Objections/Comments:		Info Segs		
		Other (Name)		
		None (Reason)		

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

SUMMARY:

Fix a number of minor bugs in unstructured and indexed files.

Unstructured files:

1. in openings for stream_input_output, a put_chars operation of zero length should truncate if not at EOF; currently acts as a no-op.
2. position (Ø,Ø) should set next byte to start of current line; currently a no-op.
3. get_chars encountering EOF should return short_record instead of end_of_info if data is returned; likewise with get_line operations.

Indexed files:

1. issuing two consecutive read operations at end of file may put the user into limbo; should just return end_of_info.
2. statistics giving count of allocated records and free blocks may be bad if zero length records have been added to the file.

Multics Change Request

TITLE: Install FAST process overseer similar to DTSS		STATUS	DATE
AUTHOR: S. Barr		Written	1 December 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>3.1</u> -Fixes Bug Number(s) _____ -Documented in MTB <u>202</u> -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 type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 12/09/75
		Expires	06/09/76
		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 Cmnd/Subr.		Document	Specify One or More
Objections/Comments: Separate MCRs for Fortran, Basic and the run unit are forthcoming.		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
SUMMARY: Install the version of the FAST process overseer that is similar to DTSS. Note: Chaining, which was described in MTB 202, is currently not planned for implementation in this release; further study on this topic is needed.		Other (Name)	FAST User Guide
		None (Reason)	
OVERVIEW: The Fast Access Subsystem for Timesharing (FAST) is an easy to use subsystem for creating and running Basic and Fortran programs. The FAST command syntax and language conventions are based on the Dartmouth Time-Sharing System (DTSS) with extensions for compatibility with Multics. Copies of the draft Users Guide to Fast can be obtained from Joan Scott at CISL.			

TITLE: Implement mail -reverse option		STATUS	DATE
AUTHOR: S. Herbst		Written	3 December 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 _____		Status	A 12/09/75
		Expires	06/09/76
		DOCUMENTATION CHANGES	
		Category (Check One)	Document Specify One or More
		<input type="checkbox"/> Lib. Maint. Tools	MPM (Vol, Sect.) Commands
		<input type="checkbox"/> Sys. Anal. Tools	PLMS (AN #)
		<input type="checkbox"/> Sys. Prog. Tools	MOSN (Sect.)
		<input type="checkbox"/> 355	MPAM (Sect.)
		<input type="checkbox"/> BOS	MSAM (Sect.)
		<input type="checkbox"/> Salvager	Info Segs
		<input type="checkbox"/> Ring Zero	Other (Name)
		<input type="checkbox"/> Ring One	None (Reason)
		<input type="checkbox"/> SysDaemon/Admin.	
		<input type="checkbox"/> Runtime	
		<input checked="" type="checkbox"/> User Cmd/Subr.	
Objections/Comments:			

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

SUMMARY:

Implement the -reverse (-rv) control argument to the mail command when reading mail. This control argument causes the messages to be printed in reverse order, latest first.

REASON:

mail's current earliest-to-latest printing is inconvenient to users who must keep old messages in their mailbox and therefore must see them again and again.

mail

mail

Name: mail, ml

The mail command allows the user to send a message to another user or to print messages in any mailbox to which he has sufficient access. The extended access used on mailboxes permits the creator of a mailbox to firmly control other users' access to his mailbox. Adding, reading, and deleting messages are independent privileges under extended access. For example, one user can be given access to only add messages, and another user to add messages and also read and delete only the messages he has added. For more information on extended access, see "Creating a Mailbox" below. Mail sent to a user is placed in the mailbox named >user_dir_dir>Project_id>Person_id>Person_id.mbx in his home directory.

Usage

To send mail:

```
mail path Person_id Project_id ... -Person_id- -Project_id-
```

where:

1. path is the pathname of a segment to be sent or is an asterisk (*) to indicate that the user wishes to type a message to be sent (see "Composing Mail" below).
2. Person_id is the name of a person to whom mail is to be sent.
3. Project_id is the name of a project on which Person_id is registered.

To print mail:

```
mail -path- -control_arg
```

where:

1. path is the pathname of a mailbox. If the mbx suffix is not given, it is assumed. If no path argument is given, the contents of the default mailbox is printed (see "Creating a Mailbox" below).
2. control_arg ~~can be~~ ~~brief,~~ ~~or~~ ~~-bf~~ ~~so that~~ Only the total number of messages in the mailbox is printed. If the mailbox is empty, nothing is printed.

-reverse, -r Messages are printed in reverse order, the most recent first. The default order is most recent last.

TITLE: Change to IO Daemon Restart Command		STATUS	DATE
AUTHOR: J. C. Whitmore		Written	11-26-75
-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 12/09/75
		Expires	06/09/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #) IO Daemon	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name) MOH, SRB	
		None (Reason)	

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

Summary:

1. Change the IO coordinator and driver (s) to make the "restart Axxx" driver command, restart all requests in series A starting at Axxx which are in the saved list, including any request which is still being processed.
2. Change the local "restart" driver command to ask the operator how many copies of a multi-copy request should be restarted.

Reasons:

1. Operators have been confused by the driver command "restart Axxx". Currently, when several requests have been lost due to printer error, the operator will "quit" the driver and give the command "Restart Axxx". However, this will not restart the request which was quit. He must also give the command "restart", but only after the printer has been fixed. If he had killed the current request after the "restart Axxx" command it would not have been restarted at all. It is confusing to describe the commands needed to do this right under the current implementation. This change would make the use of "restart Axxx" work the same under all conditions.
2. The current implementation of the local "restart" driver command (i.e., restart the current request only) will restart the request from copy number one and will charge the user for the restarted copies as well as previously completed copies. If the multi-copy request has many pages and was restarted in the middle of the last copy, the user will get some very large charges he didn't commit to pay and the printer will be tied up longer than necessary. This change will allow the operator to restart the request from the last good copy processed.

This command terminates the request that the driver is currently processing. The request is not placed in the coordinator's saved list and thus cannot be restarted later.

After completing the command, the driver looks for another request to process. (In step mode, it returns to command level.)

The restart command

Usage: restart

This command restarts the processing of the current request. ~~From the beginning.~~ It is used when the device malfunctions temporarily (e.g., when the printer runs out of paper) and part of the user's request is lost. The user is charged for ~~all completed copies, before and after the restart command.~~ *ONLY*
The requested number of EVEN THOUGH MORE COPIES ARE POSSIBLE

See "Coordinator Communication Commands" below for a description of arguments acceptable to this command.

FOR A MULTI-COPY REQUEST, THE OPERATOR IS ASKED HOW MANY OF THE COMPLETED COPIES WERE GOOD. THIS ALLOWS THE REQUEST TO BE RESTARTED FROM THE BEGINNING UP TO THE LAST GOOD COPY.

Commands That Provide Information

There are two commands that provide the operator with information to aid in the operation of the driver. These may also be used following a quit signal.

The help command

Usage: help

This command prints the name of each command that may be executed by the driver. A short description of any arguments is provided with each command name.

The status command

Usage: status

This command prints information about the current status of the driver. The information provided is:

- 1) The device name
- 2) The request type (per minor device if more than one)
- 3) Whether a request is in progress
- 4) The device status: ready, halted or not attached (if there are minor devices, this is provided per minor device)
- 5) Whether there are any pending requests
- 6) Whether step mode is set
- 7) The names of any minor devices (to be used with the ready and halt commands)

Coordinator Communication Commands

The operator must be able to prevent the loss of requests due to device malfunction. To this end, the coordinator retains each completed request in a "saved" list for a period of time to allow them to be reprocessed if needed.

The coordinator keeps track of the requests in the list by their request numbers. A request number is composed of a request series and a sequential number indicating the order in which the request was processed. For example, request number 50289 is the 289th request processed by the device within the 50000 request number series. Each device or minor device is assigned a series

of 10000 sequence numbers during initialization. The first series after coordinator initialization begins at 10001, the second series begins at 20001, and so on. This ensures that each request in the coordinator's "saved" list is uniquely identified.

There are two commands that allow the operator to control the reprocessing of requests from the driver.

The restart command

Usage: restart <request number>

This command causes requests in the coordinator's saved list to be reprocessed starting with the specified request number. Only requests in the specified request number series are restarted.

It is acceptable for one driver to restart requests in the series of another driver. However, the series must correspond to a driver of the same request type and device class as the driver executing the command. This command may be given at normal command level or at quit command level. (After a quit signal, it may also be used without an argument to restart the current driver request as described earlier.)

If the driver is restarting requests in a series that is currently assigned to a device, the series number for that device is changed by the coordinator. This includes a driver restarting requests in its own series.

A restarted request is assigned a new number when it is reprocessed. This new number becomes the request number by which the request is identified in the saved list. Therefore, if it should become necessary to restart a request a second time, the new number must be specified.

IF The request number corresponds to the series being processed by the driver (during a quit), the current request will be killed and restarted along with other requests in the series.

The save command

Usage: save <request number>

This command is used to tell the coordinator that all requests in the saved list, starting with the specified request number, are to be retained beyond the normal holding time. The action is limited to requests in the specified request number series. The save feature allows requests to be saved for possible restarting until the coordinator is logged out. Once a saved request is restarted, it is not saved any longer than the normal retention time. The coordinator never deletes the user's segment while the request is being "saved".

It is acceptable for a driver to save requests in the series of another driver. However, the specified request number must correspond to a driver of the same request type and device class as the driver executing the command.

If the specified request number series is currently assigned to a device, the series number for that device is changed by the coordinator. This includes a driver saving requests in its own series.

Commands For Terminal Control

A driver process is capable of receiving commands from two sources: 1) the normal login terminal (master terminal), and 2) a slave terminal. (See "Terminals that Control the Driver" above.)

Since the slave "terminal" can be the device itself or an additional terminal, the commands that allow the site operator (or device operator) to control the functions of the slave are separated into two categories: 1) those which apply to all slave terminals, and 2) those which only apply to an additional control terminal attached to the process.

Multics Change Request

TITLE: Hardware tape software bug fixes		STATUS	DATE	
AUTHOR: E. Stone		Written	3 December 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 3.1 -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	<input type="checkbox"/> Sys. Anal. Tools	Expires	
	<input type="checkbox"/> Sys. Prog. Tools	<input type="checkbox"/> 355	DOCUMENTATION CHANGES	
	<input type="checkbox"/> BOS	<input type="checkbox"/> Salvager	Document	Specify One or More
	<input checked="" type="checkbox"/> Ring Zero	<input type="checkbox"/> Ring One	MPM (Vol, Sect.)	
	<input type="checkbox"/> SysDaemon/Admin.	<input type="checkbox"/> Runtime	PLMS (AN #)	
	<input type="checkbox"/> User Cmmnd/Subr.		MOSN (Sect.)	
			MPAM (Sect.)	
			MSAM (Sect.)	
			Info Segs	
Objections/Comments:		Other (Name)		
		None (Reason)		

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

SUMMARY:

Fix tape_checksum_ to work for record lengths other than an integral number of words.

Fix tape_reader to accept more than one EOFs.

REASONS:

Tapes written using tape_mult_ may not boot without these changes.

Ver. 3	MULTICS CHANGE REQUEST	MCR 1451
741022		
TITLE: Auto-Call Facility		STATUS DATE
AUTHOR: Goldman		Written 09/18/75
		Status 10/21/75
		Expires 03/18/76
Planned for System: not applicable		CATEGORY (check one)
Fixes Bug Number(s): not applicable		() Lib. Maint. Tools
Documented in MTB: none		() Sys. Anal. Tools
Incompatible Changes: no		() Sys. Prog. Tools
User/Operations-visible Interface Change: no		(<input checked="" type="checkbox"/>) 355
Coded in: (<input checked="" type="checkbox"/>) PL/I () ALM (<input checked="" type="checkbox"/>) other-see below		() 80S
Performance: () better (<input checked="" type="checkbox"/>) same () worse		() Salvager
DOCUMENTATION CHANGES (specify one or more)		() Ring Zero
MPM (vol,sect) III, 15 MPAM (sect)		() Ring One
MOSN (sect) MSAM (sect)		() SysDaemon/Admin
PLMs (AN#) AN085, AN066		() Runtime
Info Segs		() User Command/Subr
Other		

OBJECTIONS/COMMENTS: Approved only as Project Overview/outline proposal. Will require 1 or more MTBs & Design Discussion describing user & system interfaces in detail, including access control, answering service interface, log interface, & definition of valid telephone #s. 1 or more MCRs must be submitted & approved before this facility can be installed.

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

SUMMARY:

Develop an Auto-Call facility, such that a normal Multics process can acquire a terminal I/O like connection to a user specified telephone number.

REASONS:

There exist Multics users who desire this facility and such a facility would allow Multics to check incoming lines to Multics automatically. Thus the telephone connections to Multics could be maintained better and more easily.

IMPLICATIONS:

This is a new facility, so no existing facilities are affected. There would be a little more work for the 355 software but only when saving the status of the port (there are additional bits to be saved) or when checking the state of the port (there is one additional possibility, i.e., start an auto-call).

Since no new interfaces are being developed, only documentation of the specific new operations will be needed. The only user level documentation required would be how to ask the answering service for an auto-call line, this would probably be in the SWG of the MPM, the internal operations would be documented in the appropriate PLM.

DETAILED PROPOSAL:

I. The Auto-Call Project

To provide the capability for a Multics process to use a telephone automatic dialing unit to dial a terminal like device (or a computer simulating a terminal) and communicating over this telephone line using normal terminal I/O routines. The acquisition of this terminal I/O channel would be dynamic, normally not require operator intervention, and be available to any process having sufficient access.

II. Software requirements

It seems that such an Auto-Call facility should be incorporated in a general way into the standard Multics software, thus

- 1) The software which actually interfaces with the telephone hardware must be updated to be able to generate an outward call and handle the resulting status data.
- 2) The software which provides the interface between user processes and the actual terminal I/O routines must be changed to provide user processes with an appropriate interface for initiating an outward call and obtaining the status of the dialing operation.
- 3) Administrative software for regulating the use of this new resource must be provided. Existing software can be used to have different connect charges (cost per hour of connect time) associated with terminal ports supporting the Auto-Call facility.

III. Implementation of the Auto-Call facility

- 1) All software for manipulating the telephone and modem hardware resides in the Honeywell 355 and so this software must be modified to support the Auto-Call hardware. The changes to this software are not particularly major, most of the new software would be additions to detect that the auto call unit should be activated, have the auto-call executed, copy the status bits in the channel status word pertaining to the Auto-Call unit (which are presently being ignored) into 355 storage and to reflect either a successful dialup or an Auto-Call failure condition back to the Multics processor. Due to the structure of the 355 software, small modifications to 355 code can have drastic effects and so careful auditing by a knowledgeable system programmer is a necessity.
- 2) The Multics hardcore and user ring tty software would be modified to support two (2) new "order" operations, one to initiate the auto call operation and one to obtain the status of the tty after the auto call attempt. Two distinct operations are needed due to the relatively long time it takes to dial a phone number. The actual phone number would be sent to the 355 using a normal write, the new order operations would use the existing "alter parameters" facility of the 6180-355 software (this facility is already used for the "wru" order call, for instance). Thus the amount of software is again not particularly major.
- 3) At the present time, all tty ports are taken by the answering service

process when Multics "comes up". When a user dials into a Multics tty port and successfully "logs in", the answering service gives the newly created user process control over the tty channel. Thus the software for having the answering service give a dialed up tty channel to a user process already exists. Note however, that only when the line is dialed up can the answering service give it away. Therefore, using another existing facility, a user process desiring to use an auto-call line would send a message to the answering service requesting that it (the answering service) obtain a line dialed to the specified phone number. The user process would also provide the answering service with an event channel and would wait for the answering service to reply. The answering service would know of any auto-call channels and use the above software (1 and 2 above) to execute the auto-call. The implementation of this new answering service facility would be very similar to the current implementation of the "wru" and "store_id" order calls. Thus, the answering service would issue the "auto call" order request and at some later point in time a wakeup on the event call channel associated with the tty port would cause the answering service to issue the "get auto call status" order call. If "auto call" order was successful (and thus the phone line is now dialed) the answering service would give the tty to the user process requesting the auto call. A wakeup on the event channel specified in the message sent to the answering service by the user process would be issued with the message containing an error code and if the dial was successful a tty port number. The user process could then attach an io switch to the indicated port and begin communication. Upon termination of the user process or a 'hangup' order call, the auto-call channel would revert to the answering service. Here again there is relatively little actual programming needed. The auto-call facility would be mostly the sequencing of existing answering service facilities.

After the code required for each of the above three (3) areas has been written, user process software to test the Auto-Call facility would also have to be written. The implementation of the Auto-Call facility requires new versions of the 355 software (which resides on the BOS tape), new hardware software (which resides on a Multics System Tape), and modifications to the online library (namely the tools library, where the answering software resides). These three separate installations would have to be tested using either the Honeywell development machine or other special system time.

IV. The Hardware

1 HSC 351 (Honeywell Auto-Call hsla board)

(Note that an hsla has room for 16 hsla boards. A normal hsla board provides the logic for two (2) hsla ports, while an Auto-Call hsla board provides only a single hsla port. That is, the logic for one hsla port requires one-half of an hsla board, the other half of the board is normally used for another port's worth of logic; however, on Auto-Call hsla boards this other half is used for the logic which operates the Auto-Call unit provided by the telephone company)

1 801A (telephone auto call unit)

1 103A modem

Multics Change Request

TITLE: Message segment improvements and bug fixes		STATUS	DATE
AUTHOR: Jerry Stern		Written	10/13/75
-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 10/21/75
		Expires	04/21/76
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	AN69
		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: Make a number of small changes to the message segment facility that accomplish the following:

1. Fix a bug in the message segment salvager that makes status permission on the parent directory of a message segment necessary for salvaging.
2. Make the syserr messages produced by the message segment salvager give more details.
3. Change the message segment facility to check for zero-length message segments before locking so that "read" and "status" type operations cannot cause unnecessary record quota overflow.

Reasons:

1. Status permission on the parent directory of a message segment should not be required for salvaging.
2. It would be useful to have more information about message segment salvages so that possible bugs could be diagnosed.
3. It is somewhat inconsistent to have a "read" or "status" type operation cause a message segment to grow.

Implications: None

Ver. 3
741022

MULTICS CHANGE REQUEST

MCR 1454

TITLE: Change append\$create_branch to use
parent access class as default
AUTHOR: L. Scheffler

STATUS	DATE
Written	10/15/75
Status	A 10/21/75
Expires	04/15/76

Planned for System: MR 3.1
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: yes
Coded in: () PL/I () ALM () other-see below
Performance: () better () same () worse
Replaces MCR: 1407

CATEGORY (check one)

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

DOCUMENTATION CHANGES (specify one or more)

MPM (vol,sect) AG93 MPAM (sect)
MOSN (sect) MSAM (sect)
PLMs (AN#)
Info Segs create_dir_changes (cd_changes)
Other

OBJECTIONS/COMMENTS:

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

SUMMARY

Add a switch "parent_ac_sw" to the create_branch_info structure (create_branch_info.incl.pl1) to indicate whether the caller has specified an access class or not. If parent_ac_sw is ON (previously MBZ), append\$create_branch will set the access class of the segment or directory being created equal to the access class of the parent directory, irrespective of whether the caller has status permission to the parent or not. If parent_ac_sw is OFF, append will use the access class provided in the info structure (this is the current operation).

Change create_dir to turn parent_ac_sw ON if the -access_class (-acc) control argument is not specified.

REASONS

Processes having directory privileges (system administrators repairing inconsistencies) can currently create segments and directories with access classes inconsistent with the access class of the parent directory unintentionally. These changes will eliminate this annoyance.

IMPLICATIONS

None. (Since parent_ac_sw has been zero, this change is upward compatible with the current create_branch_info. No new structure version is necessary.)

INFO FILES

pending_changes.info (pd_changes.info)

10/14/75 - Changes to the create_dir commands

The create_dir command has been changed to give a created directory an access class equal to the access class of the parent directory. The -access_class (-acc) control argument to create_dir may be used to create a directory with an access class different from the parent directory's access class.