TO: Distribution
FROM: Betsy Kerr
DATE: March 25, 1975
SUBJECT: Multics Change Requests

Attached are copies of Multics Change Requests which were approved from March 1 - 15, 1975.

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.
SUMMARY:
Change the command processor to attempt to determine the type of arguments expected by a command and generate the appropriate argument list or issue an error message if it can't.

REASONS:
This creates a more reasonable, consistent, efficient interface for command writers to use.

IMPLICATIONS:
1. To be done in an efficient way, we should redefine the standard entry sequence (in an upward compatible way) to determine the entry argument types in an efficient way. Currently it is necessary to get the bit count of the command segment in order to do an "object_info_" in order to get a pointer to the definitions. This can all be avoided. The details of this change are being presented in an MTB.

2. In order to call an inner-ring procedure from command level, some mechanism must be provided to determine the entry descriptor values. They can't be read from the outer ring...
DETAILED PROPOSAL:

A. The initial implementation is proposed to work as follows:

1. If a segment has a new entry format, generate the argument list as expected. However, do no conversions - only character string parameters are allowed.

2. Otherwise, generate an argument list as done today.

Note that this will require setting up a handler for "out_of_read_bracket" for gates which will be treated the old way.

B. Eventually the command processor could do conversions that are consistent with, say, PL/I.
Introducing Users to IO Daemon request types

**AUTHOR:** Jerry Stern

**SUMMARY:**
Replace all instances of the "-device_class" ("-dvc") control argument in various user commands by the "-request_type" ("-rqt") control argument. Replace the print io devices command with a new command called "print_request_types" ("prt").

**Reasons:**
Ongoing changes to the IO Daemon have introduced the new queue group concept. A queue group is a set of device classes. The device classes are now invisible to users, but are still visible to operators. Therefore, in order that consistent terminology may be used throughout the system, the current "-device_class" control argument should be replaced.

The print io devices command prints the devices that can be used for each device class, as well as the access name of the driver for each device class. The devices are not really of interest to the user. What the user really wants to know is the available queue groups, the driver access name for each, and the command (e.g., dprint) which is used for each. This information will be provided by print_request_types.

**Implications:**
The -device_class control argument, although obsolete, must still be supported for some time.

**Detailed Proposal:**
The -request_type control argument will be accepted by the following commands:

```
dprint / dpunch
list daemon requests
cancel daemon requests
pll_abs/alm_abs/fortran_abs
runoff_abs
```

See attached write-up of print_request_types
Name: print_request_types, prt

This command prints a list of all request types handled by the IO Daemon. For each request type, two items of information are provided: (1) the access name of the IO Daemon driver process which performs requests of that type, and (2) the command which should be used to submit requests of that type. An asterisk (*) immediately following a command indicates that the corresponding request type is the default for that command.

Usage

print_q_groups -control_args-

where control_args can be one or more of the following.

-brief, -bf suppresses printing of a heading line.

-access_name <an> specifies that only those queue groups having an access name of <an> are to be listed. <an> should be of the form "person.project".

-an <an> specifies that only those queue groups used by command <c> are to be listed.

-command <c> 

-cm <c>
**Title:** New administrative exec_coms

**Author:** T. Casey

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td>Written 2/26/75</td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Documentation Changes**

- **Document:** Specify One or More
- **Specify:** BOS
- **Specify:** MFM (Vol. Sect.)
- **Specify:** MPAM (Sect.)
- **Specify:** MGAM (Sect.)
- **Specify:** Info Segs
- **Specify:** Other (Name)
- **Specify:** None (Reason)

**Objections/Comments:**

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

**Summary:**

Install in >tools, new versions of the following exec_coms:

- acct_start_up.ec (asu.ec)
- admin.ec
- biller.ec
- crank.absln
- dds.absln
- err.ec
- master.ec
- system_start_up.ec
- util.ec

Rename the current versions of the above exec_coms to old.<name>.ec and install the renamed versions in >tools. The reason for this is given under IMPLICATIONS.

**Reasons**

The primary function of asu.ec is to initialize certain parts of the hierarchy, after a cold boot, at a site that is bringing up Multics for the first time. The old version of it did not work, and the initialization of a new site had to be done by hand.

During the debugging of asu.ec, certain changes to the other exec_coms mentioned above had to be made to make them operate more smoothly at a new site. In addition, a survey of the versions of these exec_coms in the tools and system administration directories at MIT and PCO revealed as many as
five different versions of the same exec_com, in various stages of obsolescence, but also containing some interesting modifications and bug fixes. Most of these (except for site-dependent modifications) have been incorporated into the official versions. (The changes involve bug fixes and checks for unusual occurrences, and should not have any compatibility implications.)

IMPLICATIONS

Bringing up a new Multics site should go more smoothly.

Billing and accounting should be more reliable.

Since each Multics site usually modifies the administrative exec_coms to suit their own needs (the modified versions being placed in the system administration directories, where they normally reside and are run from), the procedure for installing and distributing new versions of them must be formalized, and documented somewhere (possibly in SRBs). Modifications to support new features or fix bugs will be made to the official versions in tools, and submitted for installation in tools. Local system maintainers will be responsible for merging site-dependent modifications with the ones made to the official versions. To facilitate this, the old official versions will be renamed to old.<name>.ec and distributed along with the new versions. Then, compare_ascii can be used to compare the three versions of a modified exec_com, to determine which version could most conveniently be edited to incorporate the changes contained in the other.

It is possible for a site to modify these exec_coms and some related data bases in such a way that the official versions of these exec_coms will no longer work correctly at that site. The SRB should emphasize this, and advise against simple replacement of a site-modified version of an exec_com by the new official version of it.
**TITLE:** Defining an Access Class Ceiling  

**AUTHOR:** A. Kobziar

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>355</td>
<td>Document</td>
<td>Specify One or More</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface change?</th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance:</td>
<td>Better</td>
<td>Same</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Replaces MCR</th>
<th>MPM (Vol, Sect.)</th>
<th>MSAM (Sect.)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Objections/Comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info Segs</td>
</tr>
</tbody>
</table>

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

**SUMMARY:**

Add a static variable, `sys_info$access_class_ceiling` which defines the maximum access_class allowed in the system and defined in the answering service.

**REASONS:**

Currently, only 18 categories and 7 levels are defined. Since the access class field is 72 bits, a validity check against `sys_info$access_class_ceiling` will be made in `act.proc`, `reloader`, `salvager`, and `reclassify`. 
MULTICS CHANGE REQUEST

TITLE: Expand active_all_rings_data; change gm to accept -versid arg
AUTHOR: F.W. Martinson

Planned for System: MR 2.1
Fixes Bug Number(s): not applicable
Documented in MTB: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

DOCUMENTATION CHANGES (specify one or more)
HPM (vol,sect) \nMPAM (sect)
MOSN (sect) \nMSAM (sect)
PLMs (AN#)5\nPLM referring to Lib Maint Tools.
Info Segs
Other

OBJECTIONS/COMMENTS:

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

SUMMARY:
An additional field should be added to active_all_rings_data which would permit developers to have knowledge of both the Honeywell system_id running (2.1) and the MIT version running (24-7).

REASONS:
Simplifies crash analysis and debugging if developers know which version of software is running on the system.

IMPLICATIONS:
None.

DETAILED PROPOSAL:
An 8 character aci field with the name version_id will be added to active_all_rings_data.

The generate_mst command will be changed to accept a -versid control argument in the same manner as it now accepts the -sysid control argument. If only a -sysid argument is given the version_id will default to sysid. If neither a -sysid or a -versid argument are given both sysid and

Page 1
version_id will default to the system designation given to the generate_mst command. A version_id argument by itself will be valid. No header change will be required for the hardcore header since version_id processing will be set up so as to key on the existing "sysid" keyword.

NOTE: This change has already been made and tested on System M. A copy of the updated version of bound_gm_ exists in a private directory. The version of bound_gm_ under TOOLS has not been changed.
I Ver. 3
741022 MULTICS CHANGE REQUEST

TITLE: Move GCOS modules from tools to system_library_unbundled
AUTHOR: F.W. Martinson

Planned for System: MK 7.1
Fixes Bug Number(s): not applicable
Documented in MTH: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: ( )PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

Replaces MCR: Refer to MCR 833

DOCUMENTATION CHANGES (specify one or more):

MPM (vol,sect) MPAM (sect)
MSNO (sect) MSAM (sect)
PLMs (ANW)
Info Segs
Other

None (reason) not applicable

OBJECTIONS/COMMENTS:

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

SUMMARY: The following modules should be moved from system_library_tools to system_library_unbundled to complete installation of MCR 833.

bound_qcos_daemon_tls_
bound_gcos_daemon_
bound_gcos_user_
qcos_daemon_stat2_
quos_daemon_stat_
quos_abs_control

NOTE: These modules have already been moved to >system_library_unbundled and tested on system M.
**MULTICS CHANGE REQUEST**

**TITLE:** Move Language Group Info Sns to \textgreater \texttt{doc>iis}.  

**AUTHOR:** Bob May (PHX)  

Planned for System: MR 2.1  
Fixes Bug Number(s): not applicable  
Documented in MTB: not applicable  
Incompatible Change: no  
User/Operations-visible Interface Change: no  
Coded in: ( ) PL/I ( ) ALM (X) other-see below  
Performance: ( ) better (X) same ( ) worse  

**DOCUMENTATION CHANGES (specify one or more):**  
- MPX (vol,sect)  
- MOSH (sect)  
- PLMs (AN#)  
- Info Sns  
- Other: motd  

**OBJECTIONS/COMMENTS:**  

**SUMMARY:** The info sns in \textgreater \texttt{doc>iinfo} that are maintained by the Language Development group reflect the status of the compilers in the \texttt{qx} directory subtree, in addition to the installed compilers. The contents of the \textgreater \texttt{doc>iinfo} are shipped to the field as part of the Multics Releases. They should contain only info sns for the installed, supported software.

**Detailed Description:** It is proposed that the (all fortran basic) (status changes) info sns be moved to \textgreater \texttt{doc>iis}. They will be edited at time of Release to create info sns that are appropriate for the customers.
**SUMMARY:**
Modify the dprintln command to report an error and not work when a control argument follows the last pathname.

**REASONS:**
A control argument applies to all pathnames after it. A user may incorrectly assume that the position of control arguments does not matter, as is true of many other commands. This change will warn him of control arguments that do not apply to any pathname.
**TITL**: Recompile system_daily_report

**AUTHOR**: T. Casey

- **Coded in**: PL/I
- **Planned for System MR**
- **Fixes Bug Number(s)** see below
- **Documented in MFB**
- **User/Operations-visible**
  - Interface change? yes [ ] no
  - Incompatible change? yes [ ] no
  - Performance: [ ] Better [ ] Same
  - Replaces MCR

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Document Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys. Anal. Tools</td>
<td>PLMS (AN #)</td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td>MGN (Sect.)</td>
</tr>
<tr>
<td></td>
<td>MSAM (Sect.)</td>
</tr>
<tr>
<td></td>
<td>Runtime</td>
</tr>
<tr>
<td></td>
<td>User Cmmd/Subr.</td>
</tr>
</tbody>
</table>

**Objections/Comments**: Info Segs

**None (Reason)** no interface change

**SUMMARY**: CK97 number C-04718 reports a bug in system_daily_report caused by a bug in an old compiler. Recompilation fixes the bug. It has been recompiled and installed in Phoenix. This MCR is to make it official and install it at MIT, too.

The bug involved incorrect computation of several values printed by the program, because a register was assumed to be valid when it no longer was.
SUMMARY: When no pathname argument is given, ed_mgt supplies a default, but it fails to set the length of the supplied string (which `cu $arg_pbuf` would have set if an argument had been given). This causes intermittent failures. The string length will be set correctly.
<table>
<thead>
<tr>
<th>TITLE: reload directly to target segment</th>
<th>STATUS</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHOR: R. Mullen</td>
<td>Written</td>
<td>02/28/75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-Coded in:</th>
<th>Category (Check One)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X PL/I</td>
<td>Lib. Maint. Tools</td>
</tr>
<tr>
<td>ALM</td>
<td>Sys. Anal. Tools</td>
</tr>
<tr>
<td>other</td>
<td>Sys. Prog. Tools</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-Planned for System MR</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Document</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-Fixes Bug Number(s)</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specify One or More</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-Documented in MTR</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specify One or More</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-User/Operations-visible</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface change? yes</td>
<td></td>
</tr>
<tr>
<td>Incompatible change? no</td>
<td></td>
</tr>
<tr>
<td>Performance: Better</td>
<td></td>
</tr>
<tr>
<td>Replaces MCR</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 03.04.75</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expires</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/04/75</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOCUMENTATION CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
</tr>
<tr>
<td>Specify One or More</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objections/Comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info Segs</td>
</tr>
<tr>
<td>Other (Name)</td>
</tr>
<tr>
<td>None (Reason)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)</th>
</tr>
</thead>
</table>

**SUMMARY:**

It is proposed that when appropriate the reloader will read segment directly from tape into the proper place in the hierarchy, rather than reading into the process directory and then copying it.

**REASONS:**

This will be a large performance improvement for the reloader particularly in worst case situations (large segments).

**IMPLICATIONS:**

Occasionally the reloader will encounter tape errors while reading. In this case the segment will be deleted if at all possible.

**DETAILED PROPOSAL:**

1. Only to be done if called at "reload" or "i-load" entries and if debugsw is off and quota checking is inhibited.

2. All segments will be created with unique names initially. Nothing already in the hierarchy will be deleted until the segment has been successfully read from tape.
**Summary:**

Two entries in the gate `pioi_` reference nonexistent entries in ring $\varnothing$. The correct entry names should be used.

**Reasons:**

To make these gate entries functional.

**Implications:**

Operator's console on-line T&D cannot be tested until this change is made. Temporary because `pioi_` itself will be deleted when RCP is installed.

**Detailed Proposal:**

- `pioi_$consoleDetach` will be made to reference `ocdcms` $\$detach$.
- `pioi_$consoleReattach` will be made to reference `ocdcms` $\$reattach$. 
TITLE: Bring ring_zero_meter_limits_ASCII up-to-date.

AUTHOR: Paul Green

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

DOCUMENTATION CHANGES (specify one or more)
MPM (vol, sect) MPAM (sect)
MOSN (sect) MSAM (sect) X
PLMs (AN#) 52
Info Segs
Other

OBSERVATIONS/COMMENTS:

SUMMARY:
Add system_privilege_, lo1_, and pl01_ to the list of ring 0 gates accessible through ring_zero_peek_. Delete imp_dim_gate_, as it disappeared in system 24-8.

REASONS:
The list has gotten out of date.

IMPLICATIONS:
The tool meter_gate will work on all ring 0 gates again.
**TITLE:** Add -total control argument to cumulative_page_trace

**AUTHOR:** J. Gintell

<table>
<thead>
<tr>
<th>-Coded in</th>
<th>Category (Check One)</th>
<th>Status</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>I AIM</td>
<td>Sys. Anal. Tools</td>
<td>Expires</td>
<td>D 03/11/75</td>
</tr>
<tr>
<td>other</td>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**STATUS | DATE | DOCUMENTATION CHANGES**

<table>
<thead>
<tr>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>SYSTOL (Vol, Sect.) Commands</td>
</tr>
<tr>
<td></td>
<td>PLMS (Plan #)</td>
</tr>
<tr>
<td></td>
<td>MFM (Vol, Sect.)</td>
</tr>
<tr>
<td></td>
<td>SYSDAEMON/Admin.</td>
</tr>
<tr>
<td></td>
<td>MOEN (Sect.)</td>
</tr>
<tr>
<td></td>
<td>User Cmd/Subr.</td>
</tr>
<tr>
<td></td>
<td>MSAM (Sect.)</td>
</tr>
</tbody>
</table>

**Objections/Comments:**

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

**SUMMARY:** Add -total control argument to cumulative_page_trace to allow user to obtain a summary of the number of pages faulted on for each segment.

**REASONS:** Users have to perform this operation manually—the command already accumulates the information and merely must print it out.
Name: cumulative_page_trace, cpt

The cumulative_page_trace command accumulates page trace data so that the total set of pages used during the invocation of a command or subsystem can be determined. The command accumulates data from one invocation to the next. Output from the command is in tabular format showing all pages that have been referenced. A trace in the format of that produced by the page_trace command can also be obtained.

It operates by sampling and reading the system trace array after invocation of a command and at repeated intervals. Control arguments are given to specify the detailed operation of cpt.

The command line used to invoke cpt includes the command or subsystem to be traced as well as optional control arguments.

Usage

```
cumulative_page_trace (cpt) command_line -control arguments-
```

`command_line` is a character string to be interpreted by the command processor as a command line. The procedures invoked by this command line are metered by cpt. If this string contains blanks it must be surrounded by quotes.

At least one of three generic operations must be requested. They may all be combined and are performed in the following order:
- reset the table of accumulated data,
- call the command to be metered applying the specified options,
- print the results applying the various options specified.

Control Arguments

```
-control arguments

-reset, -rs reset the table of accumulated data
-flush flush core before each invocation of the command line
-loop n call the command to be metered n times
-sleep n sleep for n seconds after each call to the command to be metered
-interrupt n n VCPU milli-seconds
-int n for pagefault sampling
-timers include all faults between signal and restart
-trace path write trace on file path using io_switch named cpt.out; cpt will attach and detach this switch
-print, -pr print the accumulated results
-total, -tt print total number of faults for each segment only
-count, -ct print the accumulated results giving the number of faults for each page
-long, -lg produce output in long format, giving full pathnames
-shortline format output for a linelength of 80
```

Notes

The default mode of operation is to take no interrupts for page fault
# Multics Change Request

**TITLE:** Fix bug in status -mode  
**AUTHOR:** Steve Herbst

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
<tr>
<td>355</td>
<td></td>
</tr>
<tr>
<td>Salvager</td>
<td></td>
</tr>
<tr>
<td>Ring Zero</td>
<td></td>
</tr>
<tr>
<td>Ring One</td>
<td></td>
</tr>
<tr>
<td>SysDaemon/Admin.</td>
<td></td>
</tr>
<tr>
<td>Runtime</td>
<td></td>
</tr>
<tr>
<td>X User Cmd/Subr.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOCUMENTATION CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify One or More</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objections/Comments:</th>
</tr>
</thead>
</table>

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

**SUMMARY:**

Fix the status command so that status msf -mode reports the ring brackets of a multisegment file correctly.
### SUMMARY:

The result returned by the arc tangent routines in `bound_pll_operators` (arc_tangent_ and double_arc_tangent_) are not always correct when provided two arguments. Also, the double precision algorithm can be improved.

### REASONS:

The code for these routines was transcribed from PL/I (EPL) programs. A bug was introduced and another copied from the original. The routines failed for certain boundary conditions and input arguments indicating either of two quadrants.

### IMPLICATIONS:

The routines will return reasonable approximations for all input arguments.

### DETAILED PROPOSAL:

Correct routines have been written and tested.
**TITLE:** Improved FORTRAN I/O package

**AUTHOR:** David Levin

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
<tr>
<td>355</td>
<td></td>
</tr>
</tbody>
</table>

**CODED INTO:**
- [x] PL/I
- [ ] ALM
- [ ] other

**Planned for System MR:**

**Fixes Bug Number(s):** 120

**Documented in MTB:**

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

**Performance:** [x] Better  [ ] Same  [ ] Worse

**User/Cmd/Subr.**

**DOCUMENTATION CHANGES**

<table>
<thead>
<tr>
<th>MPM (Vol, Sect.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLMS (AN #) AN83</td>
</tr>
<tr>
<td>MOSN (Sect.)</td>
</tr>
<tr>
<td>MSAM (Sect.)</td>
</tr>
</tbody>
</table>

**Objections/Comments:**

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

**SUMMARY:**
1. Fix bug 120 and unreported bugs
2. Eliminate obsolete entries (scc, occ, imag, endfile)
3. Add set_cc to bound_fortran_io
4. Move exit into fortran_stop
5. Combine all I/O procedures into a single procedure.
6. Improve runtime error messages
7. Call default_error_handler_xlabelingadd_finish_handler once per process.

**REASONS:**
Item #5 above results in fewer full PL/I calls per I/O element. The rest are bug fixes or for consistency with SRB2.0.

**IMPLICATIONS:**
The obsolete programs are no longer available to the user. FORTRAN I/O throughput is improved. No user interface changes.
TITLE: Install a new p11 I/O runtime package.

AUTHOR: R. Schoeman

STATUS DATE
Written 03/06/75
Expires 03/11/75

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

PL/I Other
PL/I A1M
PL/I Other

Objections/Comments:

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

SUMMARY:

Install the bound_plio2_ presently in the experimental library.

REASONS:

To support the new environment (stringvalue) option, which is described on the following pages.

To support the new faster record i/o, in which the time taken for read, write, and rewrite operations on unkeyed files is reduced up to 63%.

To fix p11 bug 1324, in which the "error" condition was raised in record_directed i/o instead of the more appropriate "transmit" condition, and bug 1296, in which the p11 onfile() builtin function was inconsistent about the length of the file name.

This bound_plio2_ must be installed before the new p11_operators_ with the faster record i/o is installed.
A file may be declared or opened with the environment(stringvalue) option which then becomes an attribute of the file-state block. If a file-state block has the environment(stringvalue) attribute, the execution of read, write, and rewrite statements is affected as follows:

(read statement):
If there is an <into> option referencing a scalar varying character (or bit) string, the complete record in the file is treated as a fixed character (or bit) string value and is assigned to the variable by a normal string assignment.

(write statement) or (rewrite statement):
If there is <from> option referencing a scalar varying character (or bit) string, the record placed in the file will be a fixed length character (or bit) string that is equal to the current value of the variable.

This feature allows one to use record i/o with varying strings using only the space in the file actually needed to store the current value of the varying string.

(END)
An <entry attribute> of the form "entry()" is equivalent to an 
<entry attribute> of the form "entry", except that the former is 
a complete <attribute> and the latter is an incomplete 
<attribute>. The significance of this difference is shown in 
section 5.2.1.2 and section 5.3.2.

A <parameter descriptor list> does not restrict the values that 
may be represented by the item. A <parameter descriptor list> is 
significant only when an entry value represented by the item is 
invoked.

The <parameter descriptor list> must produce a declaration for 
each <parameter descriptor> that is equivalent to the actual 
declaration of each parameter in the entry invoked by each 
invocation of the entry values represented by this item. Such 
declarations are equivalent only if they contain exactly the same 
<attribute set>s, except that the <parameter descriptor> cannot 
have: the <parameter attribute> or <internal attribute>.

An <attribute set> of a <descriptor> must be consistent. An 
<attribute set> of a <descriptor> is consistent only if it can be 
transformed into a <descriptor set> as described in section 5.5.

A <descriptor> of a structure has exactly the same syntax as a 
<defactored declaration> of a structure variable, except that it 
has no name. Its members are declared exactly like the members 
of a structure variable, except that they have no names.

Example:

    declare F entry(1,2 fixed,2 pointer,1,
                2 bit(1),2 bit(4),(10,10) pointer);

The entry F has three parameters. The first is a structure 
containing an integer and a pointer. The second is a structure 
containing two bit-strings, and the third is a ten-by-ten array 
of pointers.

5.4.18 Environment

Syntax:

    <environment attribute>::= {environment|env}[(interactive)]

A file constant declared with an <environment attribute> causes 
the file-state block that it identifies to be opened with the 
<environment attribute>. A file-state block with an <environment 
attribute> causes the execution of each <put statement> that 
references the file to finish its output by writing a linemark. 
The <environment attribute> is normally used when the data stream 
attached to the file-state block is an interactive device used 
for both input and output.
If a completed \textit{attribute set} contains an \textit{environment attribute}, it must contain exactly one \textit{environment attribute} with a parenthesized keyword, which may be "interactive" or "stringvalue".

If a file-state block has an \textit{environment attribute} specifying "stringvalue", the execution of a \textit{read statement}, \textit{rewrite statement}, or \textit{write statement} is affected as follows. If a \textit{read statement} has an \textit{into option} referencing a scalar variable with the \textit{character attribute} and the \textit{varying attribute}, the complete record in the file is treated as a character string value and is assigned to the variable by a normal string assignment. If a \textit{rewrite statement} or \textit{write statement} has a \textit{from option} referencing a scalar variable with the \textit{character attribute} and the \textit{varying attribute}, the record placed in the file will be a character string that is equal to the current value of the variable. This form of the \textit{environment} attribute is useful for processing a file containing character strings of different lengths, especially when the file was not created by use of PL/I record output.
Required Attributes

<table>
<thead>
<tr>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>stream</td>
</tr>
<tr>
<td>input</td>
</tr>
<tr>
<td>sequential</td>
</tr>
</tbody>
</table>

4. If the filename is "sysprint" and the file control block is external and the filedescription contains the <stream attribute> and the <output attribute>, augment the filedescription with the <print attribute>.

5. The filedescription must now be a set of <attribute>s described by the following syntax:

<consistent file description>::= <stream description>;
<record description>

<stream description>::= stream{input|output[print]
[environment(interactive)]}

<record description>::= record{input|output|update}
{<sequential description>|<direct description>[
environment(string value)]}

<sequential description>::= sequential[keyed]
<direct description>::= direct keyed

6. If the filedescription contains the <print attribute> and the opening is being performed by the execution of an <open statement> and the <opening> contains a <pagesize option>, pagesize is set to the converted value of the <pagesize option>; otherwise, it is set to a default value that depends on the device or data set to which the stream is attached. If the stream is attached to a console, pagesize is set to infinity, thereby preventing an endpage condition from occurring; otherwise it is set to 60.

If the opening is being performed by the execution of an <open statement> and the <opening> contains a <pagesize option>, the filedescription must contain the <print attribute>.

7. If the filedescription contains a <stream attribute> and an <output attribute>, and the opening is being performed by the execution of an <open statement> and the <opening> contains a <linesize option>, linesize is set to the converted value of the <linesize option>; otherwise, it is set to a default value that depends on the device or data set to which the stream is attached. If the stream is attached to a console, linesize is set to the current linesize of the console; otherwise, it is set to 132.
2. If \( f \) is closed, open it as described in section 11.3. After \( f \) is open, it must have the \(<\text{input attribute}>\) or the \(<\text{update attribute}>\). If a \(<\text{key option}>\) is given, \( f \) must have the \(<\text{keyed attribute}>\) and if \( f \) has the \(<\text{direct attribute}>\), a \(<\text{key option}>\) must be given.

3. Free any input buffer associated with \( f \). This circumstance occurs when the previous input operation on \( f \) was the execution of a \(<\text{read statement}>\) containing a \(<\text{set option}>\).

4. If an \(<\text{ignore option}>\) is specified, set \( \text{currentrecord} \) to designate the \((k-1)\)th record following the record designated by \( \text{nextrecord} \). Signal the endfile condition if the value of \( k \) would position \( \text{currentrecord} \) off the end of the data set.

If a \(<\text{key option}>\) is specified, set the \( \text{currentrecord} \) of \( f \) to designate the record identified by the converted value of the \(<\text{expression}>\) in the \(<\text{key option}>\). If no such record exists in the data set attached to \( f \), signal the key condition.

If no \(<\text{key option}>\) or \(<\text{ignore option}>\) is specified, set \( \text{currentrecord} \) to the value of \( \text{nextrecord} \). If \( \text{nextrecord} \) is null, signal the endfile condition.

5. If \( f \) has the \(<\text{sequential attribute}>\), set \( \text{nextrecord} \) to designate the record following the new current record. If there is no next record, set \( \text{nextrecord} \) null.

6. If a \(<\text{keyto option}>\) is specified, assign the key associated with the current record to the variable identified by the \(<\text{reference}>\) given in the \(<\text{keyto option}>\).

7. If an \(<\text{into option}>\) is specified, assign a copy of the current record to the variable identified by the \(<\text{into option}>\). If the file-state block has an \(<\text{environment attribute}>\) specifying "stringvalue", and the variable, \( X \), referenced by the \(<\text{into option}>\) is a scalar variable with the \(<\text{character attribute}>\) and the \(<\text{varying attribute}>\) perform the assignment by executing an \(<\text{assignment statement}>\) of the form

\[ X = R; \]

where \( R \) is the record treated as character string value. If this assignment would raise the \(<\text{string size condition}>\), raise \(<\text{record condition}>\) instead. Otherwise perform the assignment by executing an \(<\text{assignment statement}>\) of the form:

\[ \text{unspec}(X) = \text{unspec}(R); \]

If \( \text{length}(\text{unspec}(R)) \neq \text{length}(\text{unspec}(X)) \), signal the record condition. On return from the \(<\text{on unit}>\), complete the assignment as if the length of \( R \) and the length of \( X \) were the minimum of the lengths of \( X \) and \( R \).

8. If a \(<\text{set option}>\) is specified, allocate a generation of storage of sufficient size to hold a copy of the current record in "system storage" and associate the generation with \( f \) as its input buffer. Assign a copy of the current record to this buffer and assign a pointer value identifying the
Constraints:

A <rewrite statement> must contain exactly one <file option> and cannot contain more than one <key option> or more than one <from option>.

Evaluation of the <reference> in the <file option> must yield a scalar file value.

Evaluation of the <key option> must yield a scalar arithmetic or string value.

Evaluation of the <from option> must yield a generation of connected storage.

Semantics:

A <rewrite statement> is executed by performing the following steps in the indicated order:

1. Evaluate the <rewrite option>s in an unspecified order.

   Let f denote the file-state block identified by the value of the <file option>.

   Convert the value of the <expression> in the <key option> to a character-string.

2. If f is closed, open it as described in section 11.3. After f is opened, it must have the <update attribute>. If a <key option> is specified, f must have the <keyed attribute>.

3. If a <key option> is specified, set the currentrecord of f to designate the record identified by the converted value of the <key option>. If no such record exists in the data set attached to f, signal the key condition.

   If a <key option> is specified and f has the <sequential attribute>, set nextrecord to designate the record following the new current record. If there is no next record, set nextrecord null.

   If no <key option> is specified, currentrecord must not be null.

4. If the file-state-block has an <environment attribute> specifying "stringvalue", a <from option> is specified, and the variable, X, referenced by the <from option> is a scalar variable with the <character attribute>, and the <varying attribute> replace the record designated by the current record with a character string equal to the current value of X.

   If a <from option> is specified, and the preceding paragraph does not apply, replace the record designated by currentrecord with a copy of the variable identified by the <reference> in the <from option>.

   If f does not have the <keyed attribute> and the size of the variable is not equal to the size of the record designated
1. Evaluate the <write option>s in an unspecified order.

Let f denote the file-state block identified by the value of the <file option>.

Convert the value of the <expression> in the <keyfrom option> to a character-string.

2. If f is closed, open it as described in section 11.3. After f is opened, it must have the <record attribute>. It cannot have the <input attribute>. If it has the <update attribute>, it must also have the <keyed attribute>. If the <keyfrom option> is specified, f must have the <keyed attribute> and if f has the <keyed attribute> the <keyfrom option> must be specified.

3. If there is an output buffer associated with f, create a new record in the data set and write the content of the buffer as the value of the new record. If there is an evaluated key associated with the buffer, associate it with the record as its key. If any record in the data set already has this key, signal the key condition.

If f has the <keyed attribute> create the new record in its proper position within the data set as determined by its key; otherwise, append the new record to the end of the data set.

After the record is written, free the output buffer. An output buffer exists when the previous output operation on f was the execution of a <locate statement>.

4. If the <keyfrom option> is specified and the data set already contains a record whose associated key is the converted value of the <keyfrom option>, signal the key condition. If currentrecord is not null, and f has both the <keyed attribute> and the <sequential attribute>, and the converted value of the <keyfrom option> is not greater than the key of the record designated by currentrecord, signal the key condition.

5. If f has the <keyed attribute> create the new record in its proper position within the data set as determined by its key; otherwise, append the new record to the end of the data set. If the variable, X, referenced by the <from option> is a scalar variable with the <character attribute> and the <varying attribute>, and if the file-state-block has an <environment attribute> specifying "stringvalue", the record is a character string equal to the current value of X. Otherwise, the record is a copy of the content of the generation identified by the evaluated <from option>.

6. Associate the converted value of the <keyfrom option> with the new record as its key.
**TITLE:** New error codes

**AUTHOR:** R. Bratt

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Status Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td>Written 03/07/75</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td>Expires 03/07/75</td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
</tbody>
</table>

**DETAILED PROPOSAL:**

Coded in error table language.

**SUMMARY:**

Two new error codes must be added to error table for use by new address space management primitives. These may as well be added to the system now.

- error_table $link link, This operation is not allowed for a link entry
- error_table $known_in_other_rings kotherrings, The segment number is being used in other rings.

**DOCUMENTATION CHANGES:**

- MPM (Vol. Sect.) error table
- PLMS (AN #)
- MOSN (Sect.)
- MSAM (Sect.)

**Objections/Comments:**

- Info Segs
- Other (Name)
- None (Reason)
SUMMARY:

When the name space management moves out of ring zero, find will have to be split into an outer ring piece and a ring zero piece. I propose that most of this split be made now. The proposed find will continue to dissect pathnames and follow links. A new module, find_entry, will handle directory locking and searching for find.

MAJOR MODIFICATIONS:

1) Place directory locking and searching code in a separate module which will remain in ring zero after find has been retrieved.

2) Create a (pathname, segno) associative memory for find and fs_get_pathname's use (the KST will not hold pathnames in the future).

3) Eliminate the seg entry points to find (and change initiate to call find_entry).

4) Remove per_system meters from find (they are not currently used and cannot be kept when find moves to the outer ring).

5) Prevent find from locking the root when it returns error_table_$root (and change the address of find to take appropriate action).
**TITLE:** Potential bugs in hardcore

**AUTHOR:** R F Mabee

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**STATUS DATE**

- **Written:** 12 Feb 75
- **Status:** O
- **Expires:** O

**Explain in DETAILED PROPOSAL**

**Planned for System MR**

**Fixes Bug Number(s)**

- N/A

**Documented in MTR**

**User/Operations-visible Interface change?**

- Yes
- No

**Incompatible change?**

- Yes
- No

**Performance:**

- Better
- Same
- Worse

**Replaces MCR**

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

**SUMMARY:** Certain bugs which make system development more difficult are remedied. Modules affected are bootstrap2, initialize_faults, get_ptrs, make_sdw, privileged_mode_ut, init_processor, plm, deact_proc, and tc_init.

**REASONS:**

1) When bootstrap2 initializes the PDS and PRDS stack headers, it sets the signal_ptr field to -21-2 because signal_ is not yet read in. Thus the PL/I environment is not entirely set up until much later when initialize_faults happens to set these pointers. If (for example) the stack header were copied in this state, the copies would retain the invalid signal_ptr. To fix this, signal_ should be moved to Collection One; presumably signal_ and condition_ can be put in the non-wired part of bound_sss_wired_.

2) Some old segment-control modules reference SDW parameters with LDAQ instructions (i.e. fixed binary (71) templates). This not only is incorrect PL/I, but also results in incorrect operation when the caller allocated the SDW by:

```
declare 1 my_sdw automatic like sdw aligned;
```

The PL/I declaration for sdw (in sdw.incl.pl1) contains only bit strings and thus may occur on any word boundary. Those modules mis-referencing their parameters should be fixed.

3) The traffic-control state meters (tc_data$.stat) are not correctly maintained. tc_init and deact_proc must update the meters. This will also contribute towards fixing the ave-queue meter.

**IMPLICATION:** Part 1 requires a change to the header and two bind files. It adds slightly to the high-water mark.
### SUMMARY:

1. Calling `syserr` at the interrupt time with a blank string as one of the arguments caused a system crash.
   
   If the argument is a blank string, `formline_` attempts to move into a zero length string with the EIS instruction `mlr` which causes a fault. Since this fault cannot be handled at the interrupt time, the system crashes.

   A trivial change to `formline_` solves the problem. If and when the CPUs are modified not to cause this fault, this change may be removed.

2. A locking bug in the IMP DIM such that the interrupt side could get in the IMP DIM without setting the IMP DIM lock.

   Fixed `imp_input_processor` to set the lock properly.

3. A bug in the processing of NCP control messages received from other sites caused hungup network connections and NCP software errors.

   Fixed `imp_input_processor.plf`.

---

<table>
<thead>
<tr>
<th>TITLE: formline_ and imp_dim bug fixes</th>
<th>STATUS</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHOR: Raj Kanodia</td>
<td>Written</td>
<td>3/4/75</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Interface change?</th>
<th>yes</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible change?</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Performance:</td>
<td>Better</td>
<td>Same</td>
</tr>
<tr>
<td>Worse</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Replaces MCR | |

**Objections/Comments:**

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

Certain types of private 4 wire high speed lines used by the M S2400 sometimes drop carrier for 1-3 seconds. This should not be treated as a hangup of the line, but as a noise burst and the tables should wait for carrier to return.

### PROPOSAL:

Fix tty_ctl to ignore the short drop in carrier and wait for its return to normal conditions. The standard GRTS error recovery procedures will prevent any data loss.

### IMPLICATIONS:

None.

### DETAILED PROPOSAL:

Coded in io_compiler language.
**Title:** 355 reliability improvement  
**Author:** M. Grady

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

**Explain in DETAILED PROPOSAL:**

**Doc. Required:**
- [X] yes  
- [ ] no

**User/Operations-visible Interface change?**  
- [ ] yes  
- [X] no

**Incompatible change?**  
- [ ] yes  
- [ ] no

**Performance:**  
- [X] Better  
- [ ] Same  
- [ ] Worse

**Replaces MCR:**

**User/Operations-visible Interface:**

**Fixes Bug Number(s):**

**Document**  
- [X] BOS  
- [ ] Salvager  
- [ ] Ring Zero  
- [ ] Ring One  
- [ ] SysDaemon/Admin.  
- [ ] Runtime  
- [ ] User Cmd/Subr.

**Specifying One or More:**
- [X] MPM (Vol, Sect.)  
- [ ] PLM (AN #)  
- [ ] MOSN (Sect.)  
- [ ] MPAM (Sect.)  
- [ ] MSAM (Sect.)

**Objections/Comments:**
- Info Segs  
- Other (Name)  
- None (Reason) not required

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

**Summary:**

The current 355 software is very sensitive to spurious line noise from any of the inactive communications channels. Any character received on an inactive (i.e. not dialed) channel caused 72 bits of status to be set to the 6180, which promptly ignores it. If too much status is generated at once, the 355 will crash because it has run out of buffers to queue the status.

**Proposal:**

Fix qlsa; tell LSLA interface routine to ignore random characters unless carrier detect is on (i.e., the line is dialed up).

**Implications:**

The 355 will be less sensitive to noise bursts and the 6180 will not waste time processing status it will ignore anyway.

**Detailed Proposal:**

Coded in 355 MAP.
TITLE: Provide new functions for BOS SAVE/RESTOR

AUTHOR: Noel I. Morris

Summary:
New functions are provided in the BOS SAVE/RESTOR mechanism to allow copying the contents of a disk pack to another disk pack, and to allow performing the normal SAVE and RESTOR of the Multics hierarchy using disk instead of tape.

Reasons:
GM wants a facility to copy a disk pack and wants faster crash recovery.

Implications:
The time needed to perform a SAVE of the Multics hierarchy can be cut by a factor of about 2.3 by using disk instead of tape.

Proposal:
Please refer to draft of MTB-5.3.3
To: Operations

From: Noel I. Morris

Date: March 6, 1975

Subject: Extensions to the BOS SAVE/RESTOR mechanism

Two new features have been added to the BOS SAVE/RESTOR mechanism. One of these is the ability to copy the contents of a disk pack to another disk pack. The other feature is the ability to use disk as an I/O device (instead of tape) for performing SAVEs and RESTOREs. In both cases, a new configuration card is required to describe the disks to be used as I/O devices. This configuration card is described in MOSN 4.3.4.

SAVE copy facility

The copy feature of the SAVE command is designed to copy the contents of a disk pack to another pack. It can be used to copy all packs in the MULT partition in order to provide a backup copy of the disk contents at a particular moment. It can also be used to copy as much as possible from a defective disk pack onto a "clean" pack. Each Multics record copied is placed in exactly the same position on the target pack. There is no copy feature in the RESTOR command. The restoration function is effected by simply mounting the copy in place of the original disk pack.

Usage

SAVE -address extent- COPY -list- -ALL-

This command is similar to the SAVE to tape, except that the list supplied is of disk drive numbers (not area numbers) instead of tape drive numbers. If the list is omitted, drives 1 and 2 will be used. As each disk pack becomes full, it will be stopped to allow the operator to mount another pack, and the next drive in the list will be used to continue the SAVE operation. The ALL option must be given if a successful salvage has not been performed prior to attempting the SAVE COPY.

SAVE to Disk

This feature of the SAVE and RESTOR commands uses disk instead of tape for performing the same functions. During the SAVE operation, Multics disk records are copied and placed in the same format as when written to tape. Then, they are written out onto
disk. The RESTOR command is able to read this disk and recover the original Multics records.

Usage

SAVE -address extent- DISK -list- -ALL-
RESTOR -address extent- DISK -list-

The disk packs specified by the list are used instead of tape to perform the SAVE or RESTOR operation.

Timing

The time to SAVE or RESTOR the contents of a single full DSU191 pack is less than 4 minutes. This is in contrast to the 10 minutes (and 2 1/2 reels) required to SAVE or RESTOR the contents of a full pack using tape.

Note

The SAVE and RESTOR commands will refuse to read or write any disk pack as an I/O device unless it is described in the PRPH DISK card. This will prevent performing a SAVE over the contents of the Multics heirarchy.

Examples

To copy the entire contents of D191 area 3 to a pack mounted on disk drive 8:

SAVE D191 AREA 3 AREA COPY 8. ALL

To make copies of all packs in the heirarchy using disk drives 8 and 9:

SAVE COPY 8. 9.

To save the Multics heirarchy onto disks using drives 8 and 9:

SAVE DISK 8. 9.

To restore the Multics heirarchy from disk using drives 8 and 9:

RESTOR DISK 8. 9.