Attached are the Change Requests that were approved during the period from February 1 - 15, 1975.
TITLE: Fix inconsistency in set_acl

AUTHOR: Steve Herbst

- Coded in [ ] PL/I [ ] AIM [ ] other—explain in DETAILED PROPOSAL
- Planned for System MR
- Fixes Bug Number(s) ________________________
- Documented in MTB 355
- User/Operations-visible [ ] yes [X] no
- Interface change? [ ] yes [X] no
- Incompatible change? [ ] yes [X] no
- Performance: [ ] Better [X] Same [ ] Worse
- Replaces MCR

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

EXPIRES: 2/25/75

TITLE: Fix inconsistency in set_acl

AUTHOR: Steve Herbst

- Coded in [X] PL/I [ ] AIM [ ] other—explain in DETAILED PROPOSAL
- Planned for System MR
- Fixes Bug Number(s) ________________________
- Documented in MTB 355
- User/Operations-visible [ ] yes [X] no
- Interface change? [ ] yes [X] no
- Incompatible change? [ ] yes [X] no
- Performance: [ ] Better [X] Same [ ] Worse
- Replaces MCR

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

EXPIRES: 2/25/75

Objections/Comments:

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

REASONS: set_acl incorrectly assumes it is to work only on segments and MSF's whenever the first access mode argument is "null", "n" or "". As a result, the following command lines do not work correctly:

```plaintext
sa dirname null Person.Project.tag
sa ** null Person1.Project1.tag1 sma Person2.Project2.tag2
```

SUMMARY: Make these cases work right. The first should set access on dirname and the second should set access on directories only.
TITLE: changes to formline_ iteration loops

AUTHOR: L. Johnson

SUMMARY: The ^ (and ^) control characters in ioa do not always seem to do the "right" thing. The proposal is to redefine slightly how iterations work. Currently, these features are not documented in the MPM. (Also ^b, ^i, and ^s are not documented.)

DETAILED PROPOSAL: Currently, a ^ will never cause an iteration to be repeated, regardless of the repetition count, if the argument list has been exhausted. The attached addition to the ioa MPM write-up explains how the new implementation will work.

Also, a bug will be fixed in ^$ (which may not do the correct thing if the string contains nested ^ (or things of the form ^^).

IMPLICATIONS: Probably small, since these features have been documented only in a help file, and not in the MPM.
Additions to the Ion MPM writer.

- The letter "i" is identical to "d". It exists because the corresponding Fortran format letter is i.

- The letter "b" assumes bit string input and converts it to character form. It may be specified in the following forms: "b", "bh", "bn.dh", or "bn.qh". The value d, when specified, is the byte size expressed in bits. It may only take on the values 1 thru 4. If d is omitted, or less than 1, 1 is assumed. If d is greater than 4, 4 is assumed. A d of 1 results in the string being outputed in binary. A d of 2 results in quarnterary (base 4) output. A d of 3 results in octal output, and a d of 4 results in hexadecimal output. If the field width, n, is omitted, the length of the string is used. If n is specified, the string will be truncated on the right, or padded on the right with spaces, which ever is appropriate.

- The letter "s" causes the next argument in the parameter list to be ignored. A "ns" will cause the next n arguments to be ignored. A "s" will do nothing.

- The letter "z" starts an iteration loop, which will be ended by a corresponding ").". An "n" specifies that the loop is to be repeated n times. A "z" specifies a semi-infinite iteration which will be repeated until the argument list is exhausted. A "0" will cause everything in the control string up to the corresponding "z" to be ignored. Iterations may be nested up to 4 deep. The exact rules under which an iteration will terminate are explained under "z".

- The letter "j" marks the end of an iteration loop, and will either terminate the iteration, or cause it to be repeated, depending on the following rules:

1. If the iteration was semi-infinite (no repeat factor specified) then it will only be repeated if there was something in the control string between the "z" and the "j" which required an argument to be processed (such as "a", "v/", etc.), and there are arguments remaining which have not been processed. If either of these conditions is not met, the loop terminates.

2. If the iteration was finite (n specified) and there was nothing in the control string between the "z" and the "j" which required an argument to be processed, the iteration will be repeated until the repetition count is exhausted. If another repetition will require an argument, the loop is repeated only if arguments remain to be processed, regardless of the value of n.
**TITLE:** Install tape_ansi_ and tape_ibm_ I/O Modules  
**AUTHOR:** Ross E. Klinger

**Planned for System:** not applicable
**Fixes Eqq Number(s):** not applicable  
**Incompatible Change:** no
**User/Operations-visible Interface Change:** yes
**Coded in:** (M)PL/I (X)ALM (X)other-see below
**Performance:** ( )better ( )same ( )worse

**DOCUMENTATION CHANGES** (specify one or more)
- MPM (vol,sect) II/10
- MOSN (sect)
- FILMS (AN#) AN57 - in preparation
- Info Seqs
- Other

**OBJECTIONS/COMMENTS:**

---

**SUMMARY:** Install the tape_ansi_ and tape_ibm_ I/O Modules in the Standard Service System (as bound_tape_ansi_).

**REASON:** Enhance support of tapes on Multics.

**IMPLICATIONS:** Providing an interchange tape facility opens whole new realms to the Multics user. In addition, such a facility is certainly a marketing "plus".

**DETAILED PROPOSAL:** Documentation available in M. D. MacLaren's office. Note: one module is coded in ALM, and another in error_table_ language.
**TITLE:** New list command

**AUTHOR:** T. Casey

**STATUS**  
- Written: 29 Jan 75
- Expires: 28 Oct 75

**DATE**
- A 02

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

**DOCUMENTATION CHANGES**
- Document
- Specify One or More

**Planned for System**
- MR

**Progs. Tools**
- Documented in MTB 125, 149

**User/Operations-visible**
- BOS
- Salvager
- MPM (Vol, Sect.) Commands
- PLMS (AN #)
- Ring Zero
- Ring One
- MOSN (Sect.)
- SysDaemon/Admin.
- Runtime
- MPAM (Sect.)
- MSAM (Sect.)

**Interface change?** Yes

**Incompatible change?** Yes

**Performance:** Better

**Replaces MCR**

**Objections/Comments:**
- Info Segs list.info
- None (Reason)

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

**SUMMARY:** Install the new list command described in MTB's 125 and 149.

**REASONS:** The current list command is deficient in many ways, and there exist several private versions of the command.

**IMPLICATIONS:**

This is an incompatible interface change. The method of specifying certain functions is changing, and one feature — the ability to list several directories at once — is being removed.

**DETAILED PROPOSAL:**

See MTB's 125 and 149; and the attached MPM writeup.
NAME: list, ls

The list command prints information about entries in a single directory. Arguments allow the user to specify the directory to be listed, the entry types, names, and dates for which information is to be printed, the amount of information to be printed for each entry, and the order in which entries are to be listed.

The default, when no arguments are given, is to list files (segments and multi-segment files) of all names, in the working directory. For each entry, the mode, records used, primary name, and any additional names, will be printed. The two entry types will be listed in the order: segments, multi-segment files; within each entry type, entries will be printed in the order in which they are found in the directory. Preceding the listing of each entry type will be a line giving the total number of entries of that type and the total records used.

USAGE: list -entry_names- -control_arguments-

1) entry_names are the (optional) names of entries to be listed. If entry names are given, only entries having at least one name matching one of the entry names will be listed. The star convention may be used in the entry names. Except when the -exclude control argument is used, the positions of the entry names (before, after, or among the control arguments) have no significance.

2) control_arguments may be chosen from the following:

-pathname dir_path
-on dir_path causes the directory dir_path to be listed, instead of the working directory.

The following control arguments are used to specify the entry types to be listed:

-segment, -sm print information about segments.

-multisegment_file, -msf print information about multi-segment files.

-file, -f print information about files (i.e., segments and multi-segment files, in that order). This is the default.

-directory, -dr print information about directories.

-branch, -br print information about branches (i.e., segments, multi-segment files, and
directories, in that order).

- link, -lk print information about links.

- all, -a print information about all entry types (segments, multi-segment files, directories, and links, in that order).

The following control arguments are used to specify whether totals information, or detailed information for each entry, or both, should be printed:

- total, -tt print only the heading lines giving total entries and total records used, for each entry type specified.

- no_header, -nhe omit the heading lines (which are printed by default when detailed information about each entry is being printed).

The default is to print totals information, plus detailed information for each entry. When detailed information is being printed, it is arranged in columns, in the order (from left to right): date-time modified, date-time used, mode, records used, names, name count. By default, only the mode, records, and names columns are printed. If the -brief control argument (see below) or any of the control arguments corresponding to these default columns is given, then this default is overridden, and only the names column, plus any columns selected explicitly by control arguments, will be printed.

The following control arguments are used to specify which columns should be printed:

- name, -nm print the names column.

- record, -rec print the records column.

- mode, -md print the mode column.

- date-time_used, -dtu print the date-time used column.

- date-time_modified, -dtm print the date-time modified column.

The following control arguments are used to specify the printing of information about multiply-named entries:

- primary, -pri print the primary name, but not the additional names, of multiply-named entries.
-match  print only those names that match one of the
given starnames.

The default is to print all of the names of multiply-named
entries.

-count, -ct   for multiply-named entries, print the total
number of names.

The following control arguments are used to decrease the amount
of information printed, or cause it to be printed in a more
compact format:

-brief, -bf  This argument has two meanings, depending on
whether or not detailed information is being
printed. If just totals information is being
printed, this argument will cause the totals
information for all selected entry types to
be abbreviated and printed on a single line.
Otherwise, it will suppress the printing of
the default columns when they are not
explicitly named in control arguments. For
example, list -dtu -brief will cause names
and date_time_used, but not mode and records,
to be printed.

-short, -sh  print link pathnames starting two spaces
after their entry names, instead of aligning
them in column 35.

The items: mode, records, and date_time_used have no meaning for
links, and there is an additional item, link_path, associated
with each link. There is no control argument to specify that
link_path should be printed. It will be printed in all cases
except the following:

1) when only totals information is being printed;
2) when the only column being printed is the names column;
3) when the -brief control argument is given.

The following control arguments specify the order in which
entries will be printed, within each entry type:

-sort sort_key sort entries according to either name,
records, mode, date_time_used, or
date_time_modified, as specified by the
sort_key, or by the default described below.

The sort_key can be one of the following keywords:
name, nm  sort entries by primary name.
record, rec  sort entries by records used (largest first).
mode, md  sort entries by mode.
date_time_used, dtu  sort entries by date_time_used (most recent first).
date_time_modified, dtm  sort entries by date_time_modified (most recent first).

If sort_key is omitted, the default column on which to sort is determined as follows: if neither date is being printed, sort by primary name; if one of the dates is being printed, sort by that date; if both dates are being printed, sort by date-time modified.

Links can only be sorted by name or date-time modified. When sorting by any other column is specified, links will remain in the order in which they are found in the directory.

If -sort is not given, entries will be printed in the order in which they are found in the directory.

-reverse, -rv  This argument is used to reverse the order in which entries are printed. If -sort is also given, the sort will be reversed. Otherwise entries will be printed in the reverse of the order in which they are found in the directory.

The following control arguments are used to specify that only a subset of all entries in the directory be listed:

-exclude entry_name  do not list any entries that have a name that matches entry_name. The star convention can be used in entry_name.

Multiple exclude entry names can be given (each immediately preceded by the -exclude control argument). They will operate together with include entry names (those not preceded by -exclude), to limit the entries listed. All entries having a name that matches any of the exclude entry names will be excluded from the listing, and from the entries that remain, all those that have a name matching any of the include entry names will be listed.
The following control arguments are used to limit the entries listed to those having date-times (either date-time modified or date-time used) before or after specified date-times. The date-time that is used in this comparison will be date-time modified, in all cases except when date-time used is being printed or sorted on, and date-time modified is neither being printed nor sorted on, in which case, date-time used will be used in this comparison.

-`from date-time`
  - do not list entries whose date-time (selected as described above) is before the specified date-time.

-`to date-time`
  - do not list entries whose date-time (selected as described above) is after the specified date-time.

If both are given, the from date-time must be earlier than the to date-time. The value after `-from` or `-to` must be a string acceptable to the `convert_date_to_binary_` subroutine, described in MPM Volume III (Subroutines). If the date-time string contains blanks, it must be enclosed in quotation marks. Note that the from and to values are date-times, and that the default time supplied by `convert_date_to_binary_`, when only a date is given, is the current time of day.

**Example 5:**

```
list -pri -ct
```

lists all files in the working directory; for multiply-named files, the primary name is printed, but not any of the secondary names, and the total number of names that the file has is printed after the primary name, on the same line. In addition to the names column, the mode and records used columns are printed. (This is the default, and it will not be mentioned in the examples below, when only the default columns are printed.)

```
list -sm *.* -ex *.pl1
```

lists all the segments in the working directory having two-component names whose second component is not "pl1".

```
list -ex *.*
```

lists all the files in the working directory having other than two-component names.
lists all files in the working directory, sorted by date-time modified. The date-time modified column is printed, in addition to the mode, records, and names columns.

    list -name -sort dtm

lists all files in the working directory, sorted by date-time modified. Only the file names are printed.

    list -sm -name -pri -nhe

lists the primary name of each segment in the working directory, without printing a heading line or other extraneous lines.

    list -mode -pri

lists the mode and primary name of each file in the working directory.
TITLE: Install record_stream_module

AUTHOR: M. Asherman

SUMMARY: To allow record i/o on unstructured data and stream i/o on sequential data.

NOTE: MTB documentation attached. This is a simplification of the original proposal in MTB-061.
This I/O module attaches an I/O switch to a target I/O switch so that record I/O operations on the attached switch are translated into stream I/O operations on the target switch, or so that stream I/O operations on the attached switch are translated into record I/O operations on the target switch. In this way a program that uses only record I/O may process unstructured files and do I/O from/to the console. Similarly a program that used only stream I/O may process some structured files.

Entries in this module are not called directly by users; rather the module is accessed through the I/O system. See the MPM Section, the Multics I/O System, for a general description of the I/O system.

Attach Description

The attach description has the following form:

```
record_stream_ switchname2 -options-
```

where:

1. `switchname2` is the name of the target I/O switch. It need not be attached when this attachment is made. If omitted, the `-target` option must be present.

2. `options` are chosen from the following to control the transformation of records into a stream of bytes and vice-versa, or to control the target attachment:
   - `-nnl` means that a record is transformed into a stream without appending a newline character.
   - `-length n` means that the stream of bytes is converted to a sequence of records each of which has length `n`.
   - `-target attach_descrip` specifies the attachment of a uniquely named target switch. This option must occur if and only if `switchname2` is omitted, and it must be the last option in the attach description, if present.

If neither `-nnl` or `-length n` occurs, lines are transformed into records after deleting trailing newlines and records into lines by appending newlines.

Opening

The attached I/O switch may be opened for stream input, stream output, sequential input, or sequential output. The implications of the opening mode are as follows (in addition to the usual implications described in the MPM Section, Multics I/O System):

October, 1973

2-1
Section, I/O System).

1. Stream input. The target I/O switch must be open for sequential input, open for sequential_input_output, or attached and closed. In the last case, it is opened for sequential input. The sequence of records read from the target switch is transformed into a stream of bytes which are transmitted to the calling program by get_line and get_chars operations. The operation read_record is used to read the records from the target switch.

2. Stream output. The target I/O switch must be open for sequential output, open for sequential_input_output, or attached and closed. In the last case, it is opened for sequential_output. The stream of bytes written to the attached switch by the put_chars operation is transformed into a sequence of records which are written to the target switch by use of the write_record operation.

3. Sequential input. The target I/O switch must be open for stream_input, open for stream_input_output, or attached and closed. In the last case, it is opened for stream_input. The stream of bytes read from the target switch is transformed into a sequence of records, which are transmitted to the calling program by read_record operations. If the attach description specifies the default line to record transformation, the get_line operation is used to read bytes from the target switch. If the attach description specifies -length, the get_chars operation is used to read bytes from the target switch.

4. Sequential output. The target I/O switch must be open for stream_output, open for stream_input_output, or attached and closed. In the last case, it is opened for stream_output. The sequence of bytes written to the attached switch by the write_record operation is transformed into a stream of bytes which are written to the target switch by use of the put_chars operation.

Transformations

The transformation from record to stream form can be described in terms of taking records from a record switch and giving bytes to a stream switch, and similarly for stream to record. Which switch is the record switch and which the stream switch depends on the opening mode as explained under "Opening." The transformation is controlled by the options in the attach description. The details are as follows. (Note that a record is a string of bytes.)

1. record to stream (default)

   A record is taken from the record switch, a newline character is appended, and the resulting string is given to the stream switch.

2. record to stream -nul

   A record is taken from the record switch and given to the stream switch without modification.
3. Stream to record (default)
   A line (string of bytes ending with a newline character) is taken from the stream switch, the newline character is deleted, and the resulting string is given to the record switch.

4. Stream to record -length n
   To form a record, n bytes are taken from the stream switch and given to the record switch as one record.

**Buffering**

The I/O module may hold data in buffers between operations when the switch is opened for stream_output or stream_input, or for sequential_input.

**Close Operation**

The I/O module closes the target switch if and only if it opened it.

**Detach Operation**

The I/O module detaches the target switch if and only if it attached it via the -target option.

**Position Operation**

Only positioning to the beginning or end of file and skipping forward are supported, except in the default sequential case which also permits backward skipping. These operations are only supported to the extent the attachment of the target I/O switch supports them.

**Control and Modes Operations**

These are supported for opened switches in the sense that they are passed along to the I/O module for the target switch.

**Error Codes**

In addition to the error codes specified in the description of iox for the various I/O operations, this I/O module returns error codes returned by the I/O.
module for the target switch.

Example

The following commands would permit sequential input operations from the user's terminal:

io_call attach sysin record_stream_ user_input
io_call open sysin sequential_input

Each record accessed through sysin corresponds to a line read through user_input, with its trailing newline character deleted.

Example

Consider a pl1 statement of the form:

open file(x) title ("record_stream_ -target vfile_foo") -opening_mode-;

The opening mode may be "stream input", "stream output", "sequential input", or "sequential output". Sequential operations on file(x) will generate stream operations on "foo" and vice-versa, with lines transformed into records without trailing newlines or records transformed into lines by appending newlines, depending upon the mode of opening.

Example

Consider the command:

io_call attach switch record_stream_ -target record_stream_ -length 100 -target vfile_foo

If "switch" is opened for stream input, "foo" must be an existing unstructured file. The effect is equivalent to that of inserting a newline after every 100 characters of "foo" referenced by set_chars, set_line or position operations through "switch".

Alternatively, "switch" may be opened for sequential output. In this case variable length records written through "switch" are given trailing newlines and restructured into 100 character records, which are then transmitted to the sequential file, "foo".

November, 1974 2-4 AG93
<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Spooling Facility for Print Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHOR:</td>
<td>Janice B. Phillipps</td>
</tr>
<tr>
<td>SOURCE:</td>
<td>(if external) e.g., &quot;User&quot;, &quot;Marketing&quot;</td>
</tr>
<tr>
<td>MIT (PDO):</td>
<td></td>
</tr>
</tbody>
</table>

### Justification

<table>
<thead>
<tr>
<th>Classification</th>
<th>Justification</th>
<th>Replaced by proposal UCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incompatible Change</td>
<td>Marketing Requirement</td>
<td>Implemented in System</td>
</tr>
<tr>
<td>Extension</td>
<td>Conformance to Standard</td>
<td></td>
</tr>
<tr>
<td>Restriction</td>
<td>Increased Consistency</td>
<td></td>
</tr>
<tr>
<td>Performance Improvement</td>
<td>Simplification</td>
<td></td>
</tr>
<tr>
<td>Reliability Improvement</td>
<td>Generalization</td>
<td></td>
</tr>
<tr>
<td>Bug Fix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Improvement</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**REASONS:** Need a method to process print requests when Multics printer(s) down.

**Summary:** MTB 117 - with the following changes:

1) Control arguments will be preceded by keywords i.e. (-files)

2) Additional control argument will be provided to specify the maximum line length of the target printer (132 for IBM printers)

**Implications:** None, new facility.
**Multics Change Request**

**TITLE:** Change format of iocb's.

**AUTHOR:** S. Webber

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-Code in PL/I [X] AIM [ ] other - explain in DETAILED PROPOSAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Planned for System MR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Fixes Bug Number(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Documented in MPR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-User/Operations-visible/not yet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface change? [ ] yes [ ] no</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Incompatible change? [ ] yes [ ] no</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Performance: [ ] Better [ ] Same [ ] Worse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Replaces MCR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**STATUS**

- Written: 1/8/75
- Expires: 8/11/75

**DOCUMENTATION CHANGES**

- Document Specify One or More
- PSO NFM (Vol, Sect.)
- Salva Ring Zero PLMS (AN #) AN57
- Ring One MOSA (Sect.)
- System Admin/MPAM (Sect.)
- Runtime MPAM (Sect.)
- User Cmm/Subr. MSAM (Sect.)

**Objections/Comments:**

- Needs an installation plan for all sites.

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

**SUMMARY:** It is proposed that the entry variables in an iocb be replaced by a packed pointer.

**REASONS:** The size of an iocb (of which usually >5 exist) will be reduced from 96 words to 48 words. (In addition, there are several other pointers in an iocb which could be made packed. If this were done the total size could be reduced to 39 words. It is not proposed here that this be done, although...)

**IMPLICATIONS:**

Every program which includes the iocb include file will have to be recompiled. In addition, any program using or setting an entry variable will have to be (trivially) changed. Since the iocb format is not yet public, now is our last chance to do this change easily. There are already some programs converted which will have to be updated.

**DETAILED PROPOSAL:**

It is proposed that due warning be given to all users, system and otherwise, so they can convert their programs to accept either format of iocb. The version number is filled in by the iox primitives that generate iocb's so this is easy to do.

At a given time a new version of bound_iox, and all system code that includes the iocb structure will be installed. This is a combined hardcore/online installation change. User programs must be converted by the time this changeover occurs.

Currently there are few, if any, user IOSIM's using iox.
| TITLE: | Install EIS version of listen and command_processor |
| STATUS | DATE |
| Written | 2/6/75 |
| Status | Expires |
| Expired | 2/11/75 |

**CATALOG**
- XPL/I
- ALIM
- Other
- Explain in DETAILED PROPOSAL

**PLANNED FOR SYSTEM MR**
- Lib. Maint. Tools
- Sys. Anal. Tools
- Sys. Prog. Tools

**FIXES BUG NUMBER(S):**
- 355

**DOCUMENTATION CHANGES**
- Document
- Specify One or More
- Salvager
- MPM (Vol. Sect.)
- Ring Zero
- PLMS (AN #)
- Ring One
- MOSN (Sect.)
- SysDaemon/Admin.
- MSAM (Sect.)
- Runtime
- MPAM (Sect.)
- User Cmd/Subr.

**Objections/Comments:**
- Info Segs
- Other (Name)
- None (Reason) none needed. Done.

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

**SUMMARY:** Install new versions of listen and command_processor that use new buffering strategies and EIS code.

**REASONS:** Performance improvement.

**IMPLICATIONS:** None.
## Multics Change Request

**TITLE:** Fix bugs in admin_util and parse_attributes_

**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</td>
<td>January 5, 1975</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td>Status</td>
<td>Expires</td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td>A 2/11/75</td>
<td>8/1/75</td>
</tr>
</tbody>
</table>

### DOCUMENTATION CHANGES

- EXPIRES: Explain in DETAIL PROPOSAL:
  - Lib. Maint. Tools
  - Sys. Anal. Tools
  - Sys. Prog. Tools

### PROPOSAL

- Fixes Bug Number(s) unreported
- Documented in MTB
- Planned for System MR 2.1
- Coded in PL/I
- Other Category (Check One)

#### Other

- User/Operations-visible Interface change: yes (X) no
- Incompatible change: yes (X) no
- Performance: Better (X) Same
- Replaces MCR

### Objections/Comments:

- None (Reason)

### SUMMARY

Fix bugs in admin_util and parse_attributes so that the default attributes for a new project, which are kept in sys_admin_data, can be successfully initialized by asu.ec and modified by a system administrator.

### REASONS

- Bug in parse_attributes prevented attributes from being set by admin_util.
- Bug in admin_util caused three attributes to be set to the reverse of the proper values.
**Title:** Add routines to process entry values  
**Author:** M. Weaver

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

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

**Summary:** Add to cu_ the entry $decade_entry_value to return the pointer components of an entry value and the entry $make_entry_value to convert an external entry pointer to an entry variable.

See attached documentation.

**Reasons:** These operations currently can only be programmed in "cheating PL/I", by structure overlays on entry variables. The first is needed by I/O modules to set the packed pointers in icbss. The second is needed by programs that get a pointer to a procedure via hcs_$make_ptr and save it as a PL/I entry value.

**Implications:** Although these routines would be coded in alm, calling them would be much more expensive than direct code. For example, calling cu_$make_entry_value would use about 27 instructions, whereas the compiler would use 4. For this reason, programmers might avoid calling them.
Entry: `cu_$decade_entry_value`
   This entry extracts the pointer components of an entry value.

Usage
   dcl cu_$decade_entry_value entry (entry,ptr,ptr);
   call cu_$decade_entry_value (entry_value,ep_ptr, env_ptr);
1) entry_value is the entry value to be decoded. (Input)
2) ep_ptr is the entry point pointer. (Output)
3) env_ptr is the environment pointer. (Output)

Entry: `cu_$make_entry_value`
   This entry constructs an entry value from its input pointer.

Usage
   dcl cu_$make_entry_value entry (ptr,entry);
   call cu_$make_entry_value (ep_ptr, entry_var);
1) ep_ptr points to an external entry point. (Input)
2) entry_var is an entry variable to be filled in. (Output)

Note
   entry_var is set to an entry value with ep_ptr as its entry point pointer and a null pointer as its environment pointer.
**TITLE:** Fix page counting error in print conversion

**JTHOR:** Noel I. Morris

- **Coded in:** [ ] PL/I [x] AIM [□] Other-
  - explain in DETAILED PROPOSAL
- **Planned for System MR**
- **Fixes Bug Number(s)**
- **Documented in MTB**
- **User/Operations-visible Interface change?** [□] yes [x] no
- **Incompatible change?** [□] yes [x] no
- **Performance:** [□] Better [x] Same [□] Worse
- **Replaces MCR**

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

<table>
<thead>
<tr>
<th>Document</th>
<th>MPM (Vol, Sect.)</th>
<th>MPM (AN #)</th>
<th>MOGN (Sect.)</th>
<th>MPAM (Sect.)</th>
<th>MSAM (Sect.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Expires:**

**Status:**

**Written:** 75.02.06

**STATUS**

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

**Objections/Comments:**

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

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

**SUMMARY:**
A bug in the printer conversion module caused page counts to be tabulated incorrectly in certain cases.

**Proposal:**
Correct the bug.
SUMMARY: 1. Install new procedure (gcos_mme_snpl) to implement MME GESNAP and the DUMP option.
   2. Change gcos_mme_fadd & gcos_mme_fcon to report a real device code.
   3. Change gcos_gein_passl to recognize label cards & implement the "endfc" option.
   4. Change gcos_mme_inos to handle 1600 bpi tapes.
   5. Fix bug in gcos_fault_proc handling of prefix vector.
   6. Change gcos_set_slave so that gcos_slave_area_seg appears in stack.
   7. Speed up ascii/gebcd conversion routines with EIS code.
   8. Minor changes to other procedures dependent on the above.

REASONS: Fix bugs

IMPLICATIONS: Better simulation of GCOS
**TITLE:** Fix Bug in Metering of IOM Interrupts  

**AUTHOR:** Noel I. Morris  

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

**Document:** Specify One or More  
- MCR  
- MTB  
- User/Operations-visible  

**Interface change?** ☑ □ yes ☑ no  
**Incompatible change?** ☑ □ yes ☑ no  
**Performance:** ☑ Better □ Same □ Worse  
**Replaces MCR**  
**Expires**  

**STATUS**  
- Written  
- Status  
- Expires  

**SUMMARY:**  
The interrupt_meters command currently prints out negative numbers for the interrupt information for "Other" (i.e. non-IOM) interrupts. This problem is caused by the fact that one IOM interrupt may result in the handling of interrupt conditions for several channels. Thus, the total of the interrupt counts for each IOM channel exceeds the total number of IOM interrupts handled.

**Proposal:**  
Modify the iom_manager to count the number of times it has been called from the interrupt interceptor. Use this figure in computing the statistics for "Other" interrupts.
**TITLE:** Improve Multics Disk DIM

**AUTHOR:** Noel I. Morris

-Coded in: [X] PL/I  [ ] ALM  [ ] other
-explain in DETAILED PROPOSAL
-Planned for System MR  2.1
-Fixes Bug Number(s)
-Documented in MTB
-User/Operations-visible
  Interface change? [X] yes  [ ] no
-Performance change?  [X] yes  [ ] no
-Replaces MCR

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

**DOCUMENTATION CHANGES**

- Fix(es)
  -Bug Number(s) Sys. Prog. Tools
  -Documented in MRB 1355 Document

- Objection(s)
  -User /Operations-visible
  -Interface change? 0 yes
  -Incompatible change? 0 yes
  -Performance change? 0 Better
  -Replacing MCR

**Objections/Comments:**

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

**SUMMARY:**

The disk DIM must be opened up in order to provide I/O Interfacier access to disk. At that time, several simple changes can be made which should result in improved disk performance.

**Proposal:**

1. A data base lock will be provided for the disk data base. It will not be necessary to have the page table lock set in order to process disk interrupts and initiate disk I/O. This lock may be set when the page table lock is already locked. The page table lock, however, may not be set by the disk DIM with the disk database lock set.

2. The disk DIM will be changed to initiate a new I/O operation before calling page$done to post a completed operation. This scheme will help avoid undue delays trying to lock the page table lock at disk interrupt time.