TO: Distribution
FROM: Joan Scott
DATE  13 May 76
RE: Multics Change Requests (Phoenix)

Enclosed are copies of Multics Change Requests from Phoenix which were approved from 1 June 1975 through 30 February 1976.
The Unstructured Removable Disk I/O Module (rdisk_) provides the capability for users to perform explicit input/output operations from/to a disk pack, treating it as a removable medium.

This fulfills a contractual commitment.

Users will now be able to access a disk in the same fashion that they can currently utilize other peripherals.

See MTB-162.
The rdisk I/O module supports I/O from/to removable disk packs. Only direct modes are supported.

Entries in this module are not called directly by users; rather, the module is accessed through the I/O system. See the "Multics Input/Output System," for a general description of the I/O system, and "File Input/Output," for a discussion of files, both in Section IV of the MPM Reference Guide.

Attach Description

The attach description has the following form:

```
rdisk_ device_id pack_id -control_args-
```

where:

1. **device_id**
   - is a character string identifying the model number of the required disk device. Currently, only the DSS191 is supported. The device_id, D191, is used for the DSS191.

2. **pack_id**
   - is a character string identifying the disk pack to be mounted.

3. **control_args**
   - may be chosen from the following:
     - **-write**
       - indicates that the disk pack is to be written. If omitted, the operator is instructed to mount the pack with the PROTECT button pressed so that writing is inhibited.
     - **-size n**
       - indicates that the value of n is to override the value of the buff_len parameter as a record size limit for the read_record operation. (See "Notes" below.)
     - **-priv**
       - indicates that the attachment is being made by a system process and that a disk drive reserved for system functions is to be assigned.

The attachment causes the specified disk pack to be mounted on a drive of the specified type.

Opening

The only opening modes supported are direct_input and direct_update. If an I/O switch attached through rdisk_ is to be opened for update, the -write control argument must occur in the attach description. This operation has no effect on the physical device.
Delete Record Operation

This operation is not supported.

Read Length Operation

This operation is not supported.

Read Record Operation

If the amount of data to be read does not terminate on a sector boundary, the excess portion of the last sector is discarded. A code of 0 is returned in this case. (See "Notes" below.)

Rewrite Record Operation

This operation is the only output operation supported. If the amount of data to be written does not terminate on a sector boundary, the remaining portion of the last sector is filled with binary zeros. A code of 0 is returned in this case. (See "Notes" below.)

Seek Key Operation

This operation returns a status code of 0 for any key that is a valid sector number. The record length returned is always 256 (current physical sector size in characters) for any valid key. The specified key must be a character string that could have been produced by editing through a PL/I picture of "(8)9". (See "Notes" below.)

Control Operation

The following orders are supported when the I/O switch is open, except for getbounds, which is supported while the switch is attached.

changepack causes the current pack to be dismounted and another pack to be mounted in its place. The info_ptr should point to a variable length character string (maximum of 32 characters) containing the identifier of the pack to be mounted.
rdisk_

getbounds causes the lowest and highest sector numbers accessible by the caller under the current modes to be returned. The info_ptr should point to a structure of the following form:

```
dcl 1 bounds,
  2 low  fixed bin(35),
  2 high fixed bin(35);
```

setsize causes the value of the record size override setting to be reset. The info_ptr should point to an aligned fixed binary(35) quantity containing the new override value.

Modes Operation

The modes operation is supported when the I/O switch is attached. The recognized modes are listed below. Each mode has a complement indicated by the circumflex character (^) that turns the mode off.

- **label, ^label** specifies that a system-defined number of sectors at the beginning of the pack are reserved for a pack label, and that a seek_key operation is to treat any key within this area as an invalid key. (The default is on.)

- **alttrak, ^alttrak** specifies that the pack has been formatted with the assignment of alternate tracks, so that a system-defined number of sectors at the end of the pack are reserved for an alternate track area. Therefore, a seek_key operation is to treat any key within that area as an invalid key. (The default is off.)

- **wrtcmp, ^wrtcmp** specifies that the write-and-compare instruction, rather than the write instruction, is used for the rewrite_record operation. This causes all data written to be read back and compared to the data as it was prior to being written. This mode should be used with discretion, since it doubles the data transfer time of every write. (The default is off.)

Write Record Operation

This operation is not supported.

Closing

The closing has no effect on the physical device.
Detaching

The detachment causes the disk pack to be dismounted.

Notes

This I/O module is a very elementary, physical-device-oriented I/O facility, providing the basic user-level interface to a disk device. All operations are performed through calls to various I/O interface (IOI) mechanisms and resource control package (RCP) entries. Certain conditions must be satisfied before a user process can make use of this facility:

1. The system must be configured with one or more disk drives available as I/O disks.
2. The user must have access to assign the disk drive with RCP and access to the IOI gates.

This I/O module allows the user to read or write a caller-specified number of characters to or from a disk pack, beginning at a caller-specified sector number. Currently, the DSS191 is the only device type supported.

The entire disk pack is treated as a keyed direct file, with keys interpreted literally as physical sector numbers. Hence, the only allowable keys are those that can be converted into fixed binary integers that fall within the range of valid sector numbers for the given disk device under the current modes, as returned by the getbounds control operation.

If an attempt is made to read or write beyond the end of the user-accessible area on disk, the code error_table_${device_end} is returned. If a defective track is encountered or if any other unrecoverable data transmission error is encountered, the code error_table_${device_parity} is returned.

The record length is specified through the buff_len parameter in the read_record operation, and through the rec_len parameter for the rewrite operation, unless overridden by a -size control argument in the attach description. (Since by definition the file consists of the entire pack, the write operation has no meaning in this I/O module.)
The following items must be considered when using this I/O module with language input/output:

1. Device Attachment and File Opening:

   a. PL/I: A file can be attached to a disk pack in PL/I by specifying the appropriate attach description in the title option of an open statement. The open statement should also specify the record and direct attributes plus either the input or update attribute, as is appropriate. After opening, the desired modes should be set, and the current sector bounds should be obtained, through direct calls to iox$_find_iocb, iox$_modes, and iox$_control. These iox_ subroutine entry points are described in Section II.

   b. FORTRAN: It is not possible to attach a file to a disk pack within FORTRAN. Here, the attachment must be made external to the FORTRAN program, e.g., through the io_call command (described in the MPM Commands) or through use of a PL/I subroutine. FORTRAN automatically opens the file with the appropriate attributes. Also, it is impossible to set modes or obtain sector bounds from within FORTRAN. This should be done through use of a PL/I subroutine prior to the first FORTRAN reference to the file.

2. Input:

   a. PL/I: The PL/I read statement with the into and key options is used to read data from a disk pack. The input record length (buff_len) is determined by the size of the variable specified in the into option. The set option should not be used. The key should be a character string containing the character representation of the desired sector number.

   b. FORTRAN: The unformatted, keyed version of the FORTRAN read statement is used. The key must be an integer, whose value is the desired sector number. In FORTRAN, buff_len has no relationship to input variable size. Hence, the -size control argument must be specified in the attach description if the disk pack is to be read through FORTRAN. The size should be set to the length of the longest expected record.
3. Output:

a. PL/I: To perform output operations to a disk pack, the PL/I rewrite statement must be used with the from and key options specified. The size of the variable referenced in the from option determines the length of the record written to disk. The key should be a character string containing the character representation of the desired sector number.

b. FORTRAN: The unformatted, keyed version of the FORTRAN write statement must be used to perform output operations to a disk pack. The size of the output record is determined by the amount of data specified in the write list. The key must be an integer whose value is the desired sector number.
MULTICS Change Request

TITLE: TOLTS 2.2 UPDATE
AUTHOR: JOHN K. RHODES

Summary: TOLTS requires updating to meet new IOI_interface for release 2.2. Also 4 problems required correction:

1. The TOLTS printer train image for the PRT300 was wrong.
2. The wrong test page was called for the entry model console.
3. A power off condition was not explicitly reported in an error message.
4. A conflict with the name of a system segment required the change of ALLOC.PL1 to TOLTS_ALLOC.PL1.
**Multics Change Request**

**TITLE:** Fix bug in the maxflow.fortran program in the TSS Fortran Library

**AUTHOR:** R. D. Lackey

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Doc. No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**STATUS** | **DATE**
--- | ---
Written | 7/3/76
Status | A 07/21/76
Expires | 07/21/76

**DOCUMENTATION CHANGES**

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

**User/Operations-visible**

- Incompatible change? [ ] yes [ ] no
- Performance: [ ] Better [ ] Same [ ] Worse
- Replaces MCR

**Objectives/Comments:**

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

**SUMMARY:** Fix maxflow.fortran to recognize the second line of input data properly.

**REASONS:** When maxflow.fortran was modified to run under Multics an incorrect format statement was included.

**IMPLICATIONS:** The maxflow.fortran program will now recognize the second line of input data properly.

**DETAILED PROPOSAL:** The correction was applied to the fortran source program maxflow.fortran.
SUMMARY:

Fix bugs in tss_fortran.info and tss_basic.info segments
### SUMMARY
Add a feature to the Backup/Reload facility to allow dumping concurrently to tape and removable disk, at the site's option. Reloading may then take place from disk at higher speed than from tape. The proposal is described in detail in MTB-172 Rev.5.

### REASONS
Backup to Disk is needed to meet a contractual commitment to GM.

---

**TABLE:**

<table>
<thead>
<tr>
<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>
</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>Replaces MCP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Objections/Comments:**
None (Reason)

**Use these headings:** SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)
**TITLE:** GCOS SIMULATOR  
**AUTHOR:** R.H. Morrison

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Class</th>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main. Tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port. Tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dev. Eng. Tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOCU_TN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Objectives/Comments:**
- Doc_UnderNrG
- Other (Name)
  - None (Reason)
  - Bug fix

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

**SUMMARY:**
Install corrected version of gcos_set_slave_

**REASONS:**
Fix a bug in the setting of BAR/NOBAR mode.

**IMPLICATIONS:**
BAR/NOBAR modes are reversed so that:
1. Out-of-bounds checks on the gcos slave segment is inoperative.
2. Several 355 tools procedures may not work properly.
**Title:** GGOS Simulator  

**Author:** R. H. Morrison

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Written</th>
<th>Status</th>
<th>Expires</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td></td>
<td>A 08/24/76</td>
<td></td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Status:** Written 7/21/75  

**Expires:** 02/04/76

**DOCUMENTATION CHANGES**

- **Document:** Specify One or More
  - MPM (Vol, Sect.) AG92-3
  - PLMS (AN #)
  - MOSN (Sect.)
  - MPAM (Sect.)
  - MSAM (Sect.)

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

**SUMMARY:** Add a new procedure "gcps_restart" to perform the function of restarting a GGOS job after a system interruption.

Install modified procedures changed to support the restart function.

**REASONS:** Fulfill the requirements of SF-7440 (730010) Task E.

**IMPLICATIONS:** Absentee GGOS jobs will automatically be restarted.

Interactive jobs will be restarted when the command "gcps" is repeated for the job.

**DETAILED PROPOSAL:** See MTB-197.
In communicating to the File and Record Control program or to the Simulator's Input/Output Supervisor.

The file code designators and their assigned file information block (FIB) are used by the Simulator for peripheral identification and management. The FIB's are similar to the Peripheral Assignment Tables (PAT's) used by native GCOS in that each contains a description of a file and each can be pointed to by one or more filecodes.

**File Names**

Temporary and output files created by the Simulator are placed in the Multics file system hierarchy. The complete pathname of each file comprises a directory name (the complete pathname of the directory containing the file) and an entry name (the equivalent of a GCOS file name).

Each directory in the hierarchy either has a quota (a limit on the total space occupied by the files it contains) or "borrows" quota from a superior directory. Quota is measured in records. One record equals one Multics page, which is 1024 words.

Jobs that produce a large amount of output or use large temporary files could fail to complete normally if there is not enough quota in the directories being used to hold the files created by the job. The Simulator allows the files to be placed in up to four different directories (three of which can be specified by the user) to help alleviate the quota problem.

The four directories are: pdir (process directory), wdir (working directory), syot_dir (SYSOUT directory), and temp_dir (temporary directory).

The pdir and wdir directories are defined by Multics. Each process is assigned a process directory when it is created (having a quota determined by the system). The current value of this quota is 512 records. Each process has a working directory, which is initially the home directory of the user but can be changed (using the change_wdir command) to any directory to which the user has access.

The syot_dir and temp_dir directories are defined by the Simulator. They are, by default, equal to wdir and pdir, respectively, but they can be set to any directories to which the user has access (using the -syot_dir and -temp_dir control arguments on the gcos command line).

Files are placed in directories on the following basis:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdir</td>
<td>gcos_slave_area_seg</td>
</tr>
<tr>
<td>syot_dir</td>
<td>All GCOS output (print, punch, and SYSOUT files)</td>
</tr>
<tr>
<td>wdir</td>
<td>Converted (ASCII or raw) copies of output files</td>
</tr>
<tr>
<td>temp_dir</td>
<td>Saved temporary GCOS files if default temp-dir is used</td>
</tr>
<tr>
<td></td>
<td>Not saved temporary GCOS files</td>
</tr>
<tr>
<td></td>
<td>Saved temporary GCOS files if temp_dir is specified by -temp_dir argument</td>
</tr>
</tbody>
</table>
Changes to GCOS Environment Simulator Manual (AN05)

The sysprint generated by the slave programs is in BCD. The execution report produced by the simulator contains both BCD and ASCII records. ASCII is used for lines that might contain Multics pathnames, since uppercase and lowercase letters are significant and should be preserved. By default, BCD SYSOUT is translated to uppercase ASCII to simulate the appearance of SYSOUT printed by native GCOS. The -lower_case control argument can be used on the gcos command line to cause BCD SYSOUT to be translated to lowercase ASCII.

SYSOUT line limits are:

<table>
<thead>
<tr>
<th>Language</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALGOL</td>
<td>10,000</td>
</tr>
<tr>
<td>CONVER</td>
<td>1,000</td>
</tr>
<tr>
<td>COBOL</td>
<td>20,000</td>
</tr>
<tr>
<td>EXECUTE</td>
<td>5,000</td>
</tr>
<tr>
<td>FILEEDIT</td>
<td>10,000</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>12,000</td>
</tr>
<tr>
<td>GO00 FORTRAN</td>
<td>12,000</td>
</tr>
<tr>
<td>GMAP</td>
<td>10,000</td>
</tr>
<tr>
<td>JOVIAL</td>
<td>10,000</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>5,000</td>
</tr>
<tr>
<td>UTILITY</td>
<td>10,000</td>
</tr>
</tbody>
</table>

The lines counted include only those supplied by the slave program, including MME GEBDRT core dumps and MME GESNAP print lines.

JOB RESTART

A GCOS job interrupted during execution by a system interruption is, by default, restarted by the simulator when a gcos command for the job is repeated in the same working directory or with the same non-default temporary directory (or both). Activity and job restart is performed as directed by the GCOS activity control card options REST/NREST and JREST/NJREST. However, as in GCOS, activity restart may not be possible depending on internal conditions at the time of the interruption (e.g., temporary files that have been released without SAVE disposition). Job restart will occur in these cases if the JREST option is in effect.

If the interrupted GCOS job was running in an absentee process, the control argument "-restart" must have been used in order for job restart to occur. This option is used by the GCOS Daemon so that jobs submitted by it are restartable.

If job restart is not wanted, regardless of the control card options, the simulator control argument "-nosave" (or "-nosy") may be used.
In Section V for the Control Cards:

<table>
<thead>
<tr>
<th>Card</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>GMAP</td>
</tr>
<tr>
<td>ALGOL</td>
<td>JOVIAL</td>
</tr>
<tr>
<td>COBOL</td>
<td>SYSEDIT</td>
</tr>
<tr>
<td>CONVER</td>
<td>UTILITY</td>
</tr>
<tr>
<td>EXECUTE</td>
<td>355 MAP</td>
</tr>
<tr>
<td>FILEDIT</td>
<td>355 SIM</td>
</tr>
<tr>
<td>FORTRAN</td>
<td></td>
</tr>
</tbody>
</table>

add to the description of card options the following:

- **NJREST** - Do not restart this job.
- **NREST** - Do not restart this job with the current activity.
- **JREST** - Enable job restart.
- **REST** - Enable activity restart.

**NOTE:** The activity restart options (NJREST, NREST, JREST, and REST) allow the user to exercise some control over the job/activity restart cycle that occurs following a system interruption.
-dpunch_options "options", queue the converted punch files for punching by the I/O daemon, but use the dpunch control arguments supplied in the options string. The -raw argument is always used for dpunch, since the converted punch files are not suitable for punching in any other mode. The explanations under -dprint_options above, regarding quotation marks and abbreviations, apply to this argument as well. (The -raw and -dpunch arguments are implied and need not be given.)

-dpno "options"
do not perform the default conversion and daemon output of print and punch files. The default is:

-dpno -dl -dpno "-dl -raw"

Since the default for each file type (print or punch) is overridden when any of the above arguments is specified for the given file type, the -hold argument is only required when one of the file types is to be left in GCOS standard system format, with no conversion or daemon output being performed.

Control arguments governing the creation of files by the simulator:

-temp_dir path, -ld path use the pathname of a directory specified by path for temporary GCOS files. By default, the process directory is used for files that are not saved, and the working directory is used for files that are saved.

-syst_dir path, -sd path use the pathname of a directory specified by path for the GCOS format copies of print, punch, and sysout files. By default, the working directory is used. (The converted copies of these files are always placed in the working directory.)

-job_id id, -id id use the job identification specified by id in the names of files created by the simulator for this job. See the GCOS Environment Simulator manual for more information on the simulator's naming of files. The id may be any character string, of up to 18 characters, to be used in file names, or it may be one of the following control arguments:

-unique use a Multics unique name as the job id. (A unique name is a 17-character string, generated by the unique_char subroutine, beginning with an exclamation point, and guaranteed to be unique within the system.)

-jd_sep, -jd use the entryname of the job deck segment as the job id. If the entryname ends with gcos, that suffix is removed from the id. This is the default.
Other control arguments:

-**brief, -bf**
  suppress the printing of all terminal output produced by the simulator except for fatal error messages. Output from the slave program is not suppressed.

-**long, -lg**
  request that, in addition to the normal terminal output, certain lines from the execution report, including the begin and end activity lines (containing PSW, etc.) also be printed on the terminal.

-**continue, -ctu**
  continue processing the job when a nonfatal error occurs. Unless the -brief control argument is given, a warning message is printed on the user's terminal. If this argument is not given, the first nonfatal error causes the job to be rejected. Nonfatal errors occur mainly in control card processing, and are described in detail in Section V of the GCOS Environment Simulator manual under "Control Cards."

-**userlib**
  enable the use of GCOS slave software libraries supplied by the user, instead of, or in addition to, the copies of the libraries installed in the system. The use of this argument is described in Section II of the GCOS Environment Simulator manual.

-**debug, -db**
  inform the simulator that: 1) it is being run interactively; 2) by a user who is familiar with the Multics debug command, and other Multics error recovery facilities; and 3) the user wishes to be given the opportunity to use these facilities to determine the cause of, and possibly correct, any error that would otherwise cause the simulation of the job to be aborted.

-**no_bar, -nobar, -nb**
  request that the simulator run the slave programs in Multics mode instead of BAR mode. This is used for debugging the simulator. Use of this argument may produce abnormal slave program behavior.

-**nosave, -nosv**
  this argument indicates to the simulator not to save job restart control data, and (therefore) not to attempt job restart after a system interruption. By default, a save data segment is created and maintained in the working directory (or the temporary directory if one is specified by -temp_dir). Job restart will be attempted when gcos is invoked again for the job (either by the absentee restart option or by an interactive user).
The gcos command invokes the GCOS Environment Simulator to run a single GCOS job, immediately, in the user's process.

**USAGE:**

`gcos job_deck_pathname -control_arguments-`

1. `job_deck_pathname` is the pathname of a file (segment or multisegment file) containing a GCOS job deck. The file may contain ASCII lines, (as produced by one of the Multics editors) representing card images, or it may be a GCOS standard system format file, containing BCD and binary card images. It is assumed to contain ASCII lines, unless the GCOS format is specified. One way of specifying GCOS format is to have the name of the file end with ".gcos". The other way is the -gcos control argument, described below.

2. `control_arguments` may be chosen from the following list. They may appear in any order, and may precede or follow the `job_deck_pathname`. The remainder of this file is over 200 lines long. The control arguments are described in the following order:

**Input:**

- `-gc`, `-gc`
- `-asci`, `-aci`
- `-no_canonicalize`, `-no_can`, `-no`
- `-truncate`, `-tnc`

**Output:**

- `-list`, `-ls`
- `-lower_case`, `-lc`
- `-dprint`, `-dp`
- `-dprint_options options`, `-dpo options`
- `-raw`
- `-dpunch`, `-dpn`
- `-dpunch_options options`, `-dpno options`
- `-hold`, `-hd`

**File creation:**

- `-temp_dir path`, `-td path`
- `-syot_dir path`, `-sd path`
- `-job_id id`, `-id id`
- `-unique`
- `-jd_segment`, `-jd`

**Other:**

- `-brief`, `-bf`
- `-long`, `-lg`
- `-continue`, `-ctu`
- `-userlib`
- `-debug`, `-db`
- `-no_bar`, `-nobar`, `-nb`
- `-nosave`, `-nosv`

If no control arguments are given, the defaults are such that the command:
\texttt{gcos} path

is equivalent to the command:

\texttt{gcos path -aci -dpo -dl -dpno "-dl -raw" -ld -jd}

Control arguments specifying the input supplied to the simulator:

\texttt{-gcos, -gc}
the input file is in GCOS standard system format, containing BGD and binary card images. Such a file could have been produced as output of a previous GCOS job, or by the \texttt{gcos_card_utility} command.

\texttt{-ascii, -aci}
the input file contains ASCII lines, as produced by one of the Multics editors. This is the default, but this argument may be used if the name of an ASCII file happens to end in ".gcos" to avoid the necessity of renaming the file.

\texttt{-no_canonicalize, -no_can, -no}
This argument may be used to save processing time, when an ASCII input file contains no tab characters, and the fields on all the card images are aligned in the columns required by GCOS. Normally, an ASCII input file may contain tabs separating the fields on each line. The process of replacing these tabs by the appropriate number of blanks to align the fields in the columns required by GCOS is known as canonicalization. Logical tab stops are known for GCOS $ control cards and for all the languages supported by the simulator. See \texttt{ANOS} for more information on canonicalization.

\texttt{-truncate, -tnc}
if any ASCII input file (the job deck file, or any $ SELECTed file) contains lines longer than 80 characters (after canonicalization), the extra characters are to be assumed to be part of comments, and discarded without warnings. If this argument is not given, the first line longer than 80 characters will cause the job to be rejected.

Control arguments specifying the disposition of output from the simulator:

\texttt{-list, -ls}
convert print files (both SYSOUT and simulated printer) from BCD to ASCII, and delete the BCD copy. (This conversion is performed by a call to the \texttt{gcos_sysprint} command for each file.)

\texttt{-lower_case, -lc}
translate alphabetic BCD characters in print files to lower case ASCII. The default is upper case.

\texttt{-dprint, -dp}
enqueue the converted print files for printing followed by deletion, by the Multics I/O Daemon. (The \texttt{-list} argument is implied and need not be given.)
-dprint_options "options", -dpo "options"
enqueue the converted print files for printing by the I/O Daemon, but
use the dprint control arguments supplied in the options string instead
of the default of "-dl". The options must be enclosed in quotation
marks if they contain blanks or other delimiter characters recognized by
the command processor. Note that the dprint command is called via
cu$_cp$ so that a user-defined abbreviation for dprint (which supplies
default heading and destination arguments, for example) will be used in
this call. (The -list and -dprint arguments are implied and need not be
given.)

-raw
convert punch files (both sysout and simulated card punch) from BCD (or
binary) to an internal format suitable for punching by the Multics I/O
Daemon in raw mode (960 bits per card image), and delete the BCD copy.
(This conversion is performed by a call to the gcos_syspunch command for
each file.)

dpunch, -dpn
enqueue the converted punch files for punching by the I/O Daemon in raw
mode, followed by deletion. (The -raw argument is implied and need not
be given.)

dpunch_options "options", -dpno "options"
enqueue the converted punch files for punching by the I/O Daemon, but
use the dpunch control arguments supplied in the options string. The
-raw argument is always used for dpunch, since the converted punch files
are not suitable for punching in any other mode. The notes under
-dprint_options above (regarding quotation marks and abbreviations),
apply to this argument as well. (The -raw and -dpunch arguments are
implied and need not be given.)

-hold, -hd
do not perform the default conversion and Daemon output of print and
punch files. The default is:
   -dpo -dl -dpno "-dl -raw"
   
Since the default for each file type (print or punch) is overridden when
any of the above arguments is specified for the given file type, the
-hold argument is only required when one of the file types is to be left
in GCOS standard system format, with no conversion or Daemon output
being performed.

Control arguments governing the creation of files by the simulator:

-temp_dir path, -td path
specifies the pathname of a directory to be used for temporary GCOS
files. By default, the process directory is used for files that are not
saved, and the working directory is used for files that are saved.

-syot_dir path, -sd path
specifies the pathname of a directory to be used for the GCOS format
copies of print, punch, and sysout files. By default, the working
directory is used. (The converted copies of these files are always
placed in the working directory.)
-job_id id, -id id
id is the job identification, to be used in the names of files created by the simulator for this job. See ANOS for more information on the simulator's naming of files. id may be any character string, of up to 18 characters, to be used in file names, or it may be one of the following control arguments:

-unique
a Multics unique name will be used as the job id. (A unique name is a 15-character string, generated by the unique_chars subroutine, beginning with an exclamation point, and guaranteed to be unique within the system.)

-jd_seg, -jd
the entry name of the job deck segment will be used as the job id. If it ends in ".gcps", that suffix will be removed from the id. This is the default.

Other control arguments:

-brief, -bf
this argument suppresses all terminal messages except for fatal error messages.

-long, -lg
this argument requests that, in addition to the normal terminal output, the begin and end activity messages (containing PSW, etc.) from the execution report be printed on the terminal.

-continue, -ctu
continue processing the job when a non-fatal error occurs. Unless the -brief control argument is given, a warning message will be printed on the user's terminal. If this argument is not given, the first non-fatal error will cause the job to be rejected. Non-fatal errors occur mainly in control card processing, and are described in detail in ANOS, Section V, under "Control Cards".

-userlib
this argument enables the use of GCOS slave software libraries supplied by the user, instead of, or in addition to, the copies of the libraries installed in the system. The use of this argument is described in Section II of ANOS.

-debug, -db
this argument indicates to the simulator that: 1) it is being run interactively; 2) by a user who is familiar with the Multics debug command, and other Multics error recovery facilities; and 3) the user wishes to be given the opportunity to use these facilities to determine the cause of, and possibly correct, any error that would otherwise cause the simulation of the job to be aborted.

-no_bar, -nobar, -nb
This argument causes the slave programs to be run in Multics mode instead of BAR mode. This allows the debug command to be used to set
breaks (MME2 instructions, which are illegal opcodes in BAR mode) in the slave program.

-nosave, -nosv
this argument indicates to the simulator not to save job restart control data, and (therefore) not to attempt job restart after a system interruption. By default, a save_data segment is created and maintained in the working directory (or the temporary directory if one is specified by -temp_dir). Job restart will be attempted when gcos is invoked again for the job (either by the absentee restart option or by an interactive user).
SUMMARY:

Modify calc desk calculator program to use EIS 59 digit precision arithmetic for calculations. Give user the option to select a result precision from 8 to 59 digits. Convert calc to ipx_.

REASON:

calc does not make use of the full capabilities of the EIS hardware.

IMPLICATIONS:

New variable "prec" . reserved for defining the result precision.
The desk calculator facility calc has been modified to use the higher precision, higher range arithmetic of the extended instruction set decimal arithmetic instructions. The user may specify the desired precision of his results (from 8 to 59 digits) by using the internal variable prec. For example, to set precision to 16 digits the line

    prec = 16

Should be entered. To obtain the current precision the value of prec may be displayed by entering:

    prec

The default precision is set to 8 digits.

Warning:

The internal arithmetic for exponentiation, sin, cos, tan, atan, ln, and log is done with double precision binary arithmetic and therefore results obtained with precision higher than 18 should be truncated to 16 digits.

The internal constants pi and e have been entered internally to 59 digits of precision.
Name: calc

The calc command provides the user with a calculator capable of evaluating arithmetic expressions with operator precedence, a set of often-used functions, and a memory that is symbolically addressable (i.e., by identifier).

Usage

calc

initiates the command. The user can then type in expressions, assignment statements, list requests, or a quit request, separated from each other by one or more newline characters. All of these operations are described below.

Expressions

Arithmetic expressions involving real values and the operands +, -, *, /, and ** (addition, subtraction, multiplication, division, and exponentiation) can be typed in. A prefix of either plus or minus is allowed. Parentheses can be used, and blanks between operators and values are ignored. Calc evaluates the expression according to rules of precedence and prints out the results.

The order of evaluation is as follows:

1. expressions within parentheses
2. function references
3. prefix +, prefix -
4. **
5. *, /
6. +, -

For example, if the user types:

2 + 3 * 4

calc responds:

= 14

Operations of the same level are processed from left to right except for the prefix plus and minus, which are processed from right to left. This means 2**3**4 is evaluated as (2**3)**4.

Numbers can be integers (123), fixed point (1.23) and floating point (1.23e+2, 1.23e2, 1.23E2, or 1230E-1). All are stored as float dec (59). A precision of 59 digits is maintained. Variables (see below) can be used in place of constants, e.g., pi * r ** 2.

Seven functions are provided: sin, cos, tan, atan, abs, ln, and log (ln is base e, log is base 10). They can be nested to any level, e.g., sin(ln(var).5*pi/180).

Assignment Statement

The value of an expression can be assigned to a variable. The name of the variable must be from one to eight characters in length and must be made up of letters (uppercase and/or lowercase) and the underscore character (_). The form is:

<variable>=<expression>

For example, the following are legal assignment statements:

x = 2

Rho = sin(2*theta)

The calc command does not print any response to assignment statements. The variables "pi" and "e" have preassigned values accurate to 59 digits.
Precision

The user may specify the desired precision of his results (from 8 to 59 digits) by using the internal variable prec. For example, to set precision to 16 digits the line

```
prec = 16
```

should be entered. To obtain the current precision the value of prec may be displayed by entering:

```
prec
```

The default precision is set to 8 digits.

Warning:

The internal arithmetic for exponentiation, sin, cos, tan, atan, ln, and log is done with double precision binary arithmetic and therefore results obtained with precisions higher than 18 should be truncated to 16 digits.

List Request

If "list" is typed, calc prints out the names and values of all the variables that have been declared so far. The value of any individual variable can be displayed by typing the name of the variable followed by a newline.

Quit Request

Typing "q" causes calc to return to the calling program, i.e., to command level.
Examples

The lines typed by the user are preceded by an exclamation mark (!).

: calc

: 2+2
    = 4

: r = 1.5

: pi*r**2
    = 7.068583

: sin(0.01)
    = 9.999832E-3

: 143e11+(12e13
        too few)

: 143e11+(12e13
        = 1.343E+14

: list

    r = 1.5
    e = 2.718282
    pi = 3.141592

: q
TITLE: Change GCOS DAEMON ABSENTEE JOBS to default restartable

AUTHOR: R. H. Morrison

- Coded in: X PL/I ALM other-
  - Planned for System MR 3.0
  - Fixes Bug Number(s) 355
  - Documented in MTF
  - User/Operations-visible
  - 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

STATUS DATE
  - Written 07/25/75
  - Status A 08/11/75
  - Expires 01/25/76

DOCUMENTATION CHANGES
  - Specify One or More
    - Document Specified

- Replaces MCR

Objections/Comments:
  - Info Segs
  - Other (Name) GCOS Simulator manual
  - None (Reason)

Use these headings:

SUMMARY:
  - Change the absentee request to specify "restartable".

REASON:
  - To support the restart function of the GCOS Simulator.

IMPLICATIONS:
  - Absentee GCOS jobs from the GCOS DAEMON will be restarted after a system interruption.

DETAILED PROPOSAL:
  - Replace gcos_queue_job.pl1.
If input is via an IMCV magnetic tape, the operator enters the command "IMCV reel_id" (if this is a nine-track tape) or "IMCV7 reel_id" (if this is a seven-track tape). This command is followed by an argument or arguments specifying the jobs on the tape to be executed.

The daemon attaches an appropriate tape drive, reads the tape, automatically detaches the tape drive, and processes the appropriate jobs.

The operator logs out the daemon by entering "logout" via the daemon console.

NOTE: Jobs on magnetic tape can be read in while the card reader is not attached.

DIRECTORY STRUCTURE

```
directory structure

```

FLOW OF INPUT AND OUTPUT

NOTE: The term "segment" refers to either a segment or a multisegment file, as needed. A segment is a collection of instructions and/or data that is associated with a particular segment name (i.e., a file).

The Simulator job stream is read into a segment named snumb.gcos in the directory input_dir. The Snumb is taken from the $ SNUMB control card. When a $ IDENT card is encountered in the input stream, the account number field is used to do a table lookup of a person/project for which the daemon will request an absentee job be submitted.

The daemon creates an absentee file in the directory gdd>Project>Person with the file name snumb.absin, where the Snumb is taken from the $ SNUMB card. When the absentee job is executed, two output files are created: snumb.absout from the absentee job and input files from the GCOS job. Both of these output files are created in the user directories. The snumb.absout file is printed via the dprint option. The job output files are printed via the dprint option or punched on punched cards via the dpunch option, as specified by the GCOS job, for snumb.gcos.

The absentee job created by the daemon is restartable. Following a system interruption, the job will be restarted as specified by the GCOS control card restart options and the Simulator control argument "-nosave".

12/74 8/75
**Ver. 3.1**  741022  MULTICS CHANGE REQUEST

**TITLE:** FIX BUG IN GCOS_CARDUTILITY

**AUTHOR:** R.H. MORRISON

Planned for System: MR 3.0
Fixes Bug Number(s): ycsU0068
Documented in MTU: not applicable
Incompatible Change: no
User/Operations-visible Interface Change: no
Coded in: (O)PL/I ( )ALM ( )other-see below
Performance: ( )better ( )same ( )worse

**DOCUMENTATION CHANGES (specify one or more)**

- MPM (vol, sect)
- MPAH (sect)
- MOSN (sect)
- MSAM (sect)
- PLMs (ANa)
- Info Seys
- Other
- ycos_card_utility.pl1  ycos_card_utility.pl1
- ycos_card_utility.info

**OBSTRUCTIONS/COMMENTS:**

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

**SUMMARY:** 
Add character count to record control word.
Add header record to file.
Correct several typo's and errors in gcu_info

**REASONS:** Conform to GCOS system standard format.

**IMPLICATIONS:** Bugs noted above will be fixed.

**DETAILED PROPOSAL:** Install the following segments:
- ycos_card_utility.pl1
- ycos_card_utility.pl1
- ycos_card_utility.info
Comparison of gcos_card_utility.info to installed version on 10/24/75 at 11:24.

A1 9/12/74
Changed to:
B1 9/15/75

A67 -truncate, -tnc I/O
Changed to:
B67 -truncate, -tc I/O

A184 -truncate, -tnc ASCII input:
Changed to:
B184 -truncate, -tc ASCII input:

A203 (media codes 1 or 2).
Changed to:
B203 or ASCII lines (media codes 1, 2, or 6).

B268 Only one of the arguments -imcv and -library may be used in an I/O specification.
B269
B270
Inserted before:
A268 -library, -lib list Input only:

A296 Multi-file reels are supported, but only as described below.
Changed to:
B298 Multi-file reels are supported, but only as described below.

A314 Only GCOS format files can be read from or written onto tape; the -gcos control argument is implied and need not be given.
A315
A316 -retain, -ret
A317 This argument specifies that the tape is to remain attached when the command finishes. It will then be available to subsequent uses of this command without requiring the operator to mount and dismount it repeatedly. This argument must be given in each use of the command except the last, to keep the tape mounted.
Changed to:
B316 -retain, -ret
B317 This argument specifies that the tape is to remain attached, and positioned at the end of the file read or written, when the command finishes. It will then be available to subsequent uses of this command without requiring the operator to mount and dismount it repeatedly.
B318 This argument must be given in each use of the command except the last, to keep the tape mounted.
(The -label argument should not be given for a retained input tape that is positioned correctly, since it causes the tape to be rewound and searched from the beginning for the specified position or file name. An output tape will remain positioned if only the file name is given, but will be rewound and re-positioned if the position is given.)

If both n and file_name are given on input, and file_name does not match Only GCOS format files can be read from or written onto tape; the -gcos control argument is implied and need not be given.

If a tape file is specified for input (or output), it must be the only Comparison finished.
Next available bug number is -- 72
Bug count is 16
eb: f - b 00068

<table>
<thead>
<tr>
<th>bnum</th>
<th>date</th>
<th>version</th>
<th>st</th>
<th>log#</th>
<th>/s</th>
<th>fixed/vers</th>
<th>resp</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccs00068</td>
<td>750777</td>
<td>240</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S. The gcc option "-gcos_ascii results in a gcos format that is not GCOS TSS system standard ascii format.
D. The character count is missing in the RCO, and the header record is incorrect.
SUMMARY:

Install info segments for the bound_gcos_tools_procedures.

REASONS:

The info segments were not previously installed.

IMPLICATIONS:

Info segments for the tools will be easily available to the maintainer of the GCOS Simulator.

DETAILED PROPOSAL:

Install the following info segments.

gcos_build_library.info
gcos_extract_module.info
gcos_library_summary.info
gcos_pull_tapefile.info
gcos_reformat_syslib.info
dump_gcos.info
**TITLE:** Modify the GCOS Simulator to track changes in restart_fault_

**AUTHOR:** R. H. Morrison

<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>08/28/75</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td>Status</td>
<td>A 09/15/75</td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td>Expires</td>
<td>02/15/76</td>
</tr>
</tbody>
</table>

**FRONTIER:** DOCUMENTATION CHANGES

**DOCUMENTATION CHANGES**

<table>
<thead>
<tr>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOS</td>
<td>MPM (Vol, Sect.)</td>
</tr>
<tr>
<td>Ring Zero</td>
<td>PLMS (AN #)</td>
</tr>
<tr>
<td>Ring One</td>
<td>MOSN (Sect.)</td>
</tr>
<tr>
<td>SysDaemon/Admin.</td>
<td>MPAM (Sect.)</td>
</tr>
<tr>
<td>Runtime</td>
<td>MSAM (Sect.)</td>
</tr>
</tbody>
</table>

**Summary:** Install modified procedures:

```
gcos_fault_processor_
gcos_mme_processor_
```

**Reasons:**

Modifications of machine conditions after a fault or MME are illegal by the current standards.

**Implications:**

GCOS programs will not be aborted after a fault or MME.
TITLE: Modify rdisk_ to use RCP

AUTHOR: J. A. Weeldreyer

MCR 1402

Page 1 of 1

Objections/Comments:

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

SUMMARY:

Modify rdisk_ to use RCP and to implement a more efficient key conversion method.

REASON:

The RCP interface is required for rdisk_ to work in MR 3.0. The new key conversion method will provide a very slight performance improvement.

IMPLICATIONS:

The already documented restriction that a key must be in a format suitable for encoding through a pic "(8)9" will be enforced.

DETAILED PROPOSAL:

1. Calls to ioi_$io_disk_attach and ioi_$io_disk_detach are replaced by calls to rcp_$attach, rcp_$check_attach, and rcp_$detach. Also, rdisk_ will no longer power down the disk drive for a pack change, but will rely on RCP to perform this function. However, rdisk_ will continue to power down the drive at detach time.

2. In key conversion, the call to cv_dec_check_ is replaced by code which results in the performance of the conversion in line.
**TITLE:** Multics COBOL Compiler  
**AUTHOR:** W. K. O'Neill (CEO-B)  

<table>
<thead>
<tr>
<th>Category (Check One)</th>
<th>Copyright Expiration Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib. Maint. Tools</td>
<td>Expires 03/03/76</td>
</tr>
<tr>
<td>Sys. Anal. Tools</td>
<td></td>
</tr>
<tr>
<td>Sys. Prog. Tools</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document</th>
<th>Specify One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPM (Vol. Sect.)</td>
<td>X</td>
</tr>
<tr>
<td>PLMS (AN #planned)</td>
<td></td>
</tr>
<tr>
<td>MOSN (Sect.)</td>
<td></td>
</tr>
<tr>
<td>MPAM (Sect.)</td>
<td></td>
</tr>
<tr>
<td>MSAM (Sect.)</td>
<td></td>
</tr>
</tbody>
</table>

**Objections/Comments:**

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

**SUMMARY:**

This fixes bugs in the COBOL compiler currently installed in Phoenix on System M.

**REASONS:**

Several serious bugs have been uncovered by the running of the ANSI-74 audit routines, by USL, and by testing in Billerica.

**IMPLICATIONS:**

1. The bugs are of sufficient importance that the changes should be installed before COBOL is released as a product.
2. No documentation changes are required.
3. No changes are required to the commands that support the COBOL object time environment.
TITLE: FIX BUG IN GCOS_DAEMON

AUTHOR: R. H. MORRISON

- Coded in X PL/I [ ] ALM [ ] other
- explain in DETAILED PROPOSAL
- Planned for System MR 3.0
- Fixes Bug Number(s) gcs00069
- Documented in MTB

Category (Check One)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Written</td>
<td>Written</td>
<td>Written</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>A 09/29/75</td>
<td>A 09/29/75</td>
<td></td>
</tr>
<tr>
<td>Expires</td>
<td>03/15/76</td>
<td>03/15/76</td>
<td></td>
</tr>
</tbody>
</table>

DOCUMENTATION CHANGES

- Fixes Bug Number(s) gcs00069
- Documented in MBB Section
- User/Operations-visible
- Planned for System MR 3.0
- Written in MTB

Objections/Comments:

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

SUMMARY:

Fix bug in gcos_read_tape_.

REASON:

Because of the bug, if the option "-all" is given for the daemon request "imcy", then the last program on the tape is not put in /dd1>GCOS<input_dir where the GCOS Simulator expects to find it.

IMPLICATIONS:

Bug will be fixed.

DETAILED PROPOSAL:

Replace gcos_read_tape_.pl1.
**Summary:**
1. Change declaration of "remaining_len" in gcos_gein_pass1_ from "fixed bin" to "fixed bin (35)".
2. Add "*1" file to FORTRAN implicit file allocations in forty_vib in gcos_control_tables_.

**Reason:**
1. gcos_gein_pass1_ does not properly handle segments with greater than 256K characters.
2. GCOS FORTRAN compilations with the "deck" option will not produce object decks or a C* file.

**Implications:**
Bugs noted above will be fixed.

**Detailed Proposal:**
Replace:
- gcos_control_tables_.alm
- gcos_gein_pass1_.pl1
SUMMARY: Fixes all known bugs in graphic system proper, adds a few new features and extensions to certain modules.

REASONS: MPRF's, TR's, local reports, user requests.

IMPLICATIONS: Cleans up known bug list

DETAILED PROPOSAL:

Bug fixes are detailed in the submitted copy of graphics_changes.info which also includes detailed descriptions of additions. This file should be inspected at Phoenix; only that copy of the file correctly documents the submitted version of the system. (Local improvements have been made since.)

Add new entrypoint, $init table, to graphic_chars, allowing for future use of multiple user- or system-defined multiple graphic character sets.

New command, list_pgs_contents (lpc); does for permanent graphic segments as "archive t" does for archives.

Change GSP tektronix_401X_ to work on dialed terminals.

Modify a few character definitions in graphic_char_table for aesthetic reasons. Programs using graphic_char_table will automatically work with the new definitions. (i.e., changes were made to the character descriptions themselves, not the structure of the character descriptions, which these programs honor.)

Rename calcomp compatible subrs (.pgs .ge) to ccs_special_symbols = to avoid naming confusion by lib. maint. personnel (requested by PHX). Users should not be using this package - also, it is currently not documented in the published GUS. If it is felt that the old name should be retained for a while, and this is OK with the library maint. (over)
personnel, it is OK with me.

Expand the segment ccs_special_symbols.pgs from the basic 14 CalComp symbols to the full 139-symbol standard set. These represent additional capability, and will present no problems to any programs who happen to be using the database. (i.e. you only get what you ask for.)

Rather than cycle this MCR back to Washington D.C., I would rather discuss any comments you might have by phone at your convenience. My number is 202-695-2478.
This file exists to keep graphics users informed of changes to the Multics Graphics System.

Information on the latest changes follows.

11/25/75 Module graphic_matrix_util was altered to fix a problem with entrypoint decode_matrix. A simple error resulted in an incorrect X rotation component being returned in some cases.

11/03/75 Subroutine graphic_compiler was replaced today to fix a problem which was causing "Unrecognized graphic effector encountered" error messages. The graphic compiler was receiving spurious "hits" on its associative memory due to its omitting to clear the memory between invocations of the compiler.

The GSP tektronix 401x was fixed to use the correct number of characters for a 1200-baud line as a default if the baud rate of the device is unavailable for any reason.

10/31/75 Many changes were installed today to both the graphic editor and the graphic system. In detail:

System PGS calcomp_compatible_subrs_.pgs has been replaced by ccs_special_symbols_.pgs. The new PGS contains 139 special CalComp symbols, which are used by camcomp_compatible_subrs_.

A new entry, graphic_chars$init_table, exists to allow the user to specify a graphic character set table other than the system default (graphic_char_table_).

The graphic_compiler was fixed to output null nodes when they are encountered in lists. Null nodes in arrays are still elided.

The graphic_dim was repaired to check for a returned error code when asking a GSP to prepare for graphic mode.

A new command, "list_pgs_contents" ("lpc") was installed. Type "help lpc" for details.

Module lsm was altered to keep future versions of the PL/I compiler happy.

Subroutine lms_fs was extensively modified to fix problems with putting and getting symbols containing subsymbols, and to eliminate the cause of a fatal process error on some structure moves.
A bug in lsm_sym_ which sometimes caused gm_examine_symtab to return a false value for the number of symbols in a graphic segment was repaired.

Command setup_graphics was modified to attempt to restore the attach table and internal state variables to a consistent state when it encounters an error.

GSP tektronix_401X_ was modified to work correctly on "dialed" (offline) terminals. In addition, a small problem with the text_4014 entry which was causing characters of the wrong size to be used was repaired.

The input parsing rules for the graphic_editor were changed slightly. Specifically, these changes affect how input lines may be terminated. (Modification was to ge_parse_ and ge_aval_.) Type "help graphic_editor" for details.

The "list" command in the graphic editor now accepts the star convention. Symbols listed are now also alphabetized. The "quit" command has the added alias "q", and checks to make sure that it is the only input on its line. The "remove" command will now remove macros also. Output from the "replay" command uses tabs instead of spaces wherever possible, and no longer prints trailing zeroes, as space-conserving measures. Changes were made to ge_environment.incl.pl1 to effect some of these improvements.

Several character descriptions in graphic_char_table_ were modified for aesthetic reasons.

09/29/75 Module calcomp_915_ was replaced today to regress from earlier changes (09/18/75) made on the basis of faulty information, and to insert a fix for the real cause of the problem described. Missing parts of plots were due to the fact that a pen_select code (implementation of "color" effector) always raises the pen and leaves it raised. (This is undocumented and is contrary to the operation of other CalComp plotters.)

09/18/75 Changes to bound_graphic_editor_ today include:

Module ge_interpret_ was modified to use tabs in its output instead of spaces, wherever possible. This should result in some storage savings in cases where structures are replayed into segments for safekeeping.
A bug which was causing strange "Symbol X undefined" messages in ge_eval was fixed. This bug occurred whenever many nested assignments caused ge_eval to grow its symbol list within a recursive invocation -- the parent invocations would lose track of the list's location.

Two fixes were inserted into GSP's today. They were:

Component tektronix_401X was changed to remove the occurrence of an annoying little dot at the upper left-hand corner of the screen which was displayed whenever the terminal was taken out of graphic mode.

Due to a bug in the CalComp 915 controller, module calcomp_915 was modified to ignore repeated requests to select a pen which is already the pen being used. The bug caused whole plot records to be ignored if the controller encountered two immediately adjacent requests to "pen select" the same pen.

08/15/75 The graphic editor was replaced today to write around an extremely hairy PL/I compiler problem which was causing certain arcane recursions to behave entirely incorrectly. The symptom which this fix repairs was the returning of "Node out of bounds" errors on replays.

08/14/75 Several small problems with the graphic editor were repaired today. In particular:

Module ge_eval was modified to strip quotes off datablocks before creating them. Module ge_interpret was similarly modified so that quotes are printed correctly when datablocks are "shown" or "replayed".

Module ge_parse was modified to circumvent a bug in the PL/I optimizer.

Fixes to bound_graphic_system today include:

Correction of lsm_sym to prevent new graphic symbols whose names are superstrings of existing graphic symbols from clobbering the existing symbol;

Modification of lsm_fs to correct a problem with "putting" and "getting" symbols containing arrays of arrays;
A change to graphic_manipulator_add_element which assures that the element to be added is added at the desired point (after, not before, the indicated node.)

08/11/75 The graphic editor was changed today to incorporate new features and to repair a few bugs.

The "replay" command now optionally takes the control argument "-all" (or "-a"). This control argument (which must be the only argument, if given) causes "replay" to replay all user-defined symbols. Also, "replay" now places semicolons after each replayed symbol requested (if more than one symbol is requested per invocation) instead of just the last symbol. In addition, "replay" has been modified to suppress multiple replaying of the same symbol within one command invocation.

The syntax "x....5", where the number of periods given exceeds the actual number of levels in the structure, is now correctly diagnosed and rejected.

Invocation of the "remove" command will no longer cause spurious "malformed macro" diagnostics to be generated after the next line of input.

A few explicit allocations were made into implicit storage allocations, for speed.

08/08/75 Two small changes were made to bound_graphic_system today.

Command setup_graphics (and remove_graphics) were changed to save and restore the modes on switch "user_i/o" when it is necessary to detach and reattach it. Although MCS saves most of the modes across detaches, this was done to assist the Network and user-specific IO modules.

IO module graphic_dim was modified to prevent illegal modifier faults caused by assignation of an unset pointer in the attach entry.

Entries below this point document changes to the graphic system which were released in system MR3.0
Objections/Comments:

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

SUMMARY: Fixes all known bugs in graphic editor, cleans up confusing input line parsing rules, adds new features as described below.

REASONS: User requests, local MPRF's

IMPLICATIONS: Cleans up known bug list, eases use of package.

DETAILED PROPOSAL:

Bug fixes are detailed in the submitted copy of graphics_changes.info which also includes detailed descriptions of additions. This file should be inspected at Phoenix; only that copy of the file correctly documents the submitted version of the system. (Local improvements have been made since.)

The input parsing rules for the graphic editor have been made more consistent and more intuitive. Details are available in graphic_editor.info seg. Specifically, the graphic editor now assumes less (is a bit smarter) about when a newline (rather than a semicolon) is meant to end an input line. All syntaxes which previously worked will also work in the new editor; but some syntaxes which would have resulted in an implicit termination (and an error message) in the old editor will now be accepted, and valid results produced.

Module ge_interpret was changed to elide trailing zeroes in components and to use tabs in output formatting to save space when its output is directed to a file (as well as time... to a terminal.)

The list command in the graphic editor was changed to sort its output and to honor the star convention.

The replay command takes a control argument, "-all", which causes the entire contents of the working graphic segment (the "buffer") to be replayed, to allow the user to dump all his graphic structures in an understandable ASCII source form to a file for study or safekeeping. (over)
Rather than cycle this MCR back to Washington D.C., I would rather discuss any comments you might have by phone at your convenience. My number is 202-695-2478.
MULTICS CHANGE REQUEST

TITLE: Modify rdisk to implement raw mode and 3 new control orders to support disk pack formatting operations

AUTHOR: James A. Bush MSS Phoenix

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

CLASSIFICATION | JUSTIFICATION
---|---
Incompatible | Replaced by proposal MCR
Change | Implemented in System
Extension | Objections/Comments:
Restriction | Addition to current documentation to be part of Subsystem Writers' Guide.
Improvement | Performance
Reliability
Improvement | Generalization
| (Unreported)
| Bug Fix

Use these headings: REASONS, SUMMARY, IMPLICATIONS, and optionally DETAIL PROPOSAL

REASONS:
1. Implement a raw mode that will allow a user to access the entire disk pack, including the T & D cylinder which takes a special seek command.
2. Implement a device_info control order that will allow a user to get access to device specific data (sect per track, sect per device, etc) without the user having to know what type of disk device he is using.
3. Implement a rd_trk_header control order that will allow a user to obtain data relating to the specific condition of a track, (ie, if the track was formatted defective, alternate, or good).
4. Implement a format_trk control order that will allow a user to specify the track condition that is wished to be formatted, the condition of the header bypass switch and if the track is to be formatted alternate or defective, the respective defective or alternate track address.

SUMMARY: The modifications described above have already been implemented and checked out on System M. Updated documentation for rdisk is included with this MCR.

IMPLICATIONS: Current users of rdisk will not be affected.
The rdisk_ I/O module supports I/O from/to removable disk packs. Only direct modes are supported.

Entries in this module are not called directly by users; rather, the module is accessed through the I/O system. See the "Multics Input/Output System," for a general description of the I/O system, and "File Input/Output," for a discussion of files, both in Section IV of the MPM Reference Guide.

**Attach Description**

The attach description has the following form:

```
rdisk_ device_id pack_id -control_args-
```

where:

1. **device_id** is a character string identifying the model number of the required disk device. Currently, only the DSS191 is supported. The device_id, D191, is used for the DSS191.

2. **pack_id** is a character string identifying the disk pack to be mounted.

3. **control_args** may be chosen from the following: may occur only once.
   - **write** indicates that the disk pack is to be written. If omitted, the operator is instructed to mount the pack with the PROTECT button pressed so that writing is inhibited.
rdisk_

-size n  indicates that the value of n is to override the value of the buff_len parameter as a record size limit for the read_record operation. (See "Notes" below.)

-priv  indicates that the attachment is being made by a system process and that a disk drive reserved for system functions is to be assigned.

The attachment causes the specified disk pack to be mounted on a drive of the specified type.

Opening

The only opening modes supported are direct_input and direct_update. If an I/O switch attached through rdisk_ is to be opened for update, the -write control argument must occur in the attach description. This operation has no effect on the physical device.

Delete_Record_Operation

This operation is not supported.

Read_Length_Operation

This operation is not supported.
Read_RecordOperation

If the amount of data to be read does not terminate on a sector boundary, the excess portion of the last sector is discarded. A code of 0 is returned in this case. (See "Notes" below.)

Rewrite_RecordOperation

This operation is the only output operation supported. If the amount of data to be written does not terminate on a sector boundary, the remaining portion of the last sector is filled with binary zeros. A code of 0 is returned in this case. (See "Notes" below.)

Seek_KeyOperation

This operation returns a status code of 0 for any key that is a valid sector number. The record length returned is always 256 (current physical sector size in characters) for any valid key. The specified key must be a character string that could have been produced by editing through a PL/I picture of "(8)9". (See "Notes" below.)

ControlOperation

The following orders are supported when the I/O switch is open, except for getbounds, which is supported while the switch is attached.
rdisk_

changepack causes the current pack to be dismounted and another pack to be mounted in its place. The info_ptr should point to a variable length character string (maximum of 32 characters) containing the identifier of the pack to be mounted.

goingbounds causes the lowest and highest sector numbers accessible by the caller under the current modes to be returned. The info_ptr should point to a structure of the following form:

dcl 1 bounds,
  2 low fixed bin(35),
  2 high fixed bin(35);

setsize causes the value of the record size override setting to be reset. The info_ptr should point to an aligned fixed binary(35) quantity containing the new override value.

device_info causes information pertaining to the attached disk device to be returned to the user. The info_ptr should point to a structure of the following form:

dcl 1 device_info_table aligned,
  2 subsystem_name char (4),
  2 device_name char (8),
  2 sect_per_dev fixed bin (35),
  2 cyl_per_dev fixed bin,
  2 sect_per_cyl fixed bin,
  2 sect_per_track fixed bin,
  2 num_label_sect fixed bin,
  2 num_alt_sect fixed bin,
  2 sector_size fixed bin (12);

where:
1. subsystem_name is the name of the Disk subsystem in use (i.e. "D191").

2. device_name is the name of the disk device in use (i.e. "disk_04").

3. sect_per_dev is the total number of non-T&D sectors on the disk pack.

4. cyl_per_dev is the total number of non-T&D cylinders on the disk pack.

5. sect_per_cyl is the number of data sectors on each cylinder of a disk pack.

6. sect_per_track is the number of data sectors on each track.

7. num_label_sect is the number of data sectors to reserve for label information.

8. num_alt_sect is the number of data sectors to reserve for alternate track area.

9. sect_size is the number of 36 bit words in each data sector.

format_trk causes a format track command to be issued to the track that was indicated by a preceding seek_key operation. The info_ptr should point to a user supplied structure of the following form:
dcl 1 format_trk_info aligned,
   (2 hz bit (2),
   2 ti bit (2),
   2 adcyl fixed bin (16),
   2 adhd fixed bin (16)) unaligned;

where:

1. hz is a bit pattern indicating
   the state of the header bypass
   switch. The hz bits are defined
   as follows:

   h z  bit pattern meaning
   0 0 format home address
      and all data records
   0 1 verify home address
      and record one, format
      home address and all
      data records
   1 0 skip home address,
      format all data
      records
   1 1 verify home address
      and data record one,
      skip home address and
      format all data
      records

2. ti is a bit pattern indicating
   the state of the track indicator
   bits. The ti bits are defined as
   follows:

   t i  bit pattern meaning
   0 0 format trk good
   0 1 format trk alternate
   1 0 format trk defective
      with alternate trk
      assigned
   1 1 format trk defective
      with no alternate trk
assigned

3. adcyl and adhd are the alternate or defective cylinder and head numbers used when the track indicator bits equal "01"b or "10"b. These two fields are defined as follows:

If the track indicator bits are set to "01"b (alternate trk), then adcyl and adhd should be equal to the defective cylinder and head number for which the alternate track is being formatted.

If the track indicator bits are set to "10"b (defective with alternate assigned), then adcyl and adhd should be equal to the cylinder and head number of the alternate track.

rd_trk_header causes a read track header command to be issued to the track that was indicated by a preceding seek_key operation. The raw track header information is passed to the user in a structure (pointed to by info_ptr) of the following form:

dcl 1 trk_header_info aligned,
(2 ha_cyl bit (16),
 2 ha_head bit (16),
 2 pad1 bit (2),
 2 ha_ti bit (2),
 2 pad2 bit (10),
 2 rcd_O_ti bit (2),
 2 rcd_O_cyl bit (16),
 2 rcd_O_head bit (16),
 2 rcd_O_dn bit (8),
 2 pad3 bit (24),

2 rcd_0_data (8), bit (8),
2 pad4 bit (4) unaligned;

where:

1. ha_cyl is the cylinder number read from the track home address.

2. ha_head is the head number read from the track home address.

3. ha_ti is the track indicator bits (defined above in the format_trk order) read from the track home address.

4. rcd_0_ti is the track indicator bits read from record zero. If the ha_ti bits indicate "10"b, then rcd_0_ti should equal "01"b for alternate track. If ha_ti indicates "01"b, then rcd_0_ti should equal "10"b for defective track. Otherwise rcd_0_ti will equal ha_ti.

5. rcd_0_cyl and rcd_0_head are the cylinder and head number read from record zero. If ha_ti indicates "10"b, then rcd_0_cyl and rcd_0_head will equal the cylinder and head number of the alternate track. If ha_ti indicates "01"b, then rcd_0_cyl and rcd_0_head will contain the cylinder and head number of the defective track. Otherwise rcd_0_cyl and rcd_0_head will equal ha_cyl and ha_head.
6. rcd_0_rn is the record number for record zero (normally equal to zero).

7. rcd_0_data is the eight data bytes in record zero (not a normal data record) and will normally be equal to zero.

8. padn are unused bits which will be returned as "0"b.

Modes Operation

The modes operation is supported when the I/O switch is attached. The recognized modes are listed below. Each mode has a complement indicated by the circumflex character (^) that turns the mode off.

- label, ^label specifies that a system-defined number of sectors at the beginning of the pack are reserved for a pack label, and that a seek_key operation is to treat any key within this area as an invalid key. (The default is on.)

- alttrk, ^alttrk specifies that the pack has been formatted with the assignment of alternate tracks, so that a system-defined number of sectors at the end of the pack are reserved for an alternate track area. Therefore, a seek_key operation is to treat any key within that area as an invalid key. (The default is off.)

- wrtcmp, ^wrtcmp specifies that the write-and-compare instruction, rather than the write
instruction, is used for the rewrite_record operation. This causes all data written to be read back and compared to the data as it was prior to being written. This mode should be used with discretion, since it doubles the data transfer time of every write. (The default is off.)

raw, "raw

specifies that the entire disk pack is available to the user, including the T & D cylinder (the last cylinder on the disk pack). (The default is off.)

Write_Record_Operation

This operation is not supported.

Closing

The closing has no effect on the physical device.

Detaching

The detachment causes the disk pack to be dismounted.

Notes

This I/O module is a very elementary, physical-device-oriented I/O facility, providing the basic
user-level interface to a disk device. All operations are performed through calls to various I/O interfa<er (I0I) mechanisms and resource control package (RCP) entries. Certain conditions must be satisfied before a user process can make use of this facility:

1. The system must be configured with one or more disk drives available as I/O disks.
2. The user must have access to assign the disk drive with RCP and access to the I0I gates.

This I/O module allows the user to read or write a caller-specified number of characters to or from a disk pack, beginning at a caller-specified sector number. Currently, the DSS191 is the only device type supported.

The entire disk pack is treated as a keyed direct file, with keys interpreted literally as physical sector numbers. Hence, the only allowable keys are those that can be converted into fixed binary integers that fall within the range of valid sector numbers for the given disk device under the current modes, as returned by the getbounds control operation.

If an attempt is made to read or write beyond the end of the user-accessible area on disk, the code error_table_Sdevice_end is returned. If a defective track is encountered or if any other unrecoverable data transmission error is encountered, the code error_table_Sdevice_parity is returned.

The record length is specified through the buff_len parameter in the read_record operation, and through the rec_len parameter for the rewrite operation, unless overridden by a -size control argument in the attach description. (Since by definition the file consists of the entire pack, the write operation has no meaning in this I/O module.)
The following items must be considered when using this I/O module with language input/output:

1. Device Attachment and File Opening:

   a. PL/I: A file can be attached to a disk pack in PL/I by specifying the appropriate attach description in the title option of an open statement. The open statement should also specify the record and direct attributes plus either the input or update attribute, as is appropriate. After opening, the desired modes should be set, and the current sector bounds should be obtained through direct calls to iox_Sfind_iocb, iox_Smodes, and iox_Scontrol. These iox subroutine entry points are described in Section II.

   b. FORTRAN: It is not possible to attach a file to a disk pack within FORTRAN. Here, the attachment must be made external to the FORTRAN program, e.g., through the io_call command (described in the MPM Commands) or through use of a PL/I subroutine. FORTRAN automatically opens the file with the appropriate attributes. Also, it is impossible to set modes or obtain sector bounds from within FORTRAN. This should be done through use of a PL/I subroutine prior to the first FORTRAN reference to the file.

2. Input:

   a. PL/I: The PL/I read statement with the into and key options is used to read data from a disk pack. The input record length (buff_len) is determined by the size of the variable specified in the into option. The set option should not be used. The
key should be a character string containing the character representation of the desired sector number.

b. FORTRAN: The unformatted, keyed version of the FORTRAN read statement is used. The key must be an integer, whose value is the desired sector number. In FORTRAN, buff_len has no relationship to input variable size. Hence, the-size control argument must be specified in the attach description if the disk pack is to be read through FORTRAN. The size should be set to the length of the longest expected record.

3. Output:

a. PL/I: To perform output operations to a disk pack, the PL/I rewrite statement must be used with the from and key options specified. The size of the variable referenced in the from option determines the length of the record written to disk. The key should be a character string containing the character representation of the desired sector number.

b. FORTRAN: The unformatted, keyed version of the FORTRAN write statement must be used to perform output operations to a disk pack. The size of the output record is determined by the amount of data specified in the write list. The key must be an integer whose value is the desired sector number.
<table>
<thead>
<tr>
<th>Objections/Comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended installation in tools - Documentation in System Tools PLM.</td>
</tr>
</tbody>
</table>

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

REASONS:

Operations has needed a facility for formatting disk packs under Multics for some time. At present time all disk pack formatting must be done under the control of BOS or offline T&D. This implies that this utility function must be done when the system is not up and running Multics.

SUMMARY:

A disk pack formatting facility, called format_disk_pack, has been implemented and tested on System M. Documentation is included with this MCR.

IMPLICATIONS:

Operations will now be able to perform the utility function of disk pack formatting online under Multics.

This is a resubmittion of an earlier MCR which was postponed.
The `format_disk_pack` command provides a means of formatting disk packs, on line under Multics. Operations will no longer have to resort to BOS to format system packs. A removable spindle must be available in the configuration to allow this command to operate.

**Usage**

```
format_disk_pack vol_id operation -control_args
```

where:

1. **vol_id**
   - is the volume name of the disk pack to be formatted.

2. **operation**
   - is one of the following operations to be performed on the specified disk pack:

   **read_pack**
   - In this mode of operation, each track header of a disk pack is read. Data is accumulated on all tracks that are formatted "defective". After all disk pack headers are read, a summary of the defective tracks is displayed on the user's terminal. This operation may be performed either interactively or absentee.

   **format_pack**
   - In this mode of operation, each track header is read and the track condition is interrogated. A format_track command is then issued, using the information from the previous track condition in the following manner:

     If the track had previously been formatted "good" or "alternate", then the track is reformatted "good".

     If the track had previously been formatted as "defective alternate track assigned" or "defective no alternate track assigned", then the track is reformatted "defective no alternate track assigned" (if the "-nodef" control argument was not specified).
3. control_args

   -nodef

   can be one or more of the following control arguments.

   will force all tracks to be formatted as "good" regardless of the previous track condition. This control argument will be in effect only for the format_pack operation.
format_disk_pack

-hold will allow the user (interactive user only) to go into an option loop after the current operation is complete so that he may select another operation without releasing the disk pack. The user may exit from the option loop by typing "quit" instead of the operations described above.

-list will produce a segment in the user's working directory with the name of "volume_id.pad_trks". This segment will contain an array of the binary seek address's of all defective tracks found on the current disk pack. This segment may be defined by the following pli structure declaration:

dcl 1 pad_trks based (bad_trk_ptr),
    2 count fixed bin
    2 def_trks fixed bin(35);(,xx)
The sort command was first released with Multics Release 3.0. Additional functions proposed for Multics Release 4.0 are:

1. A subroutine interface for the Sort is added (sort).

2. A Merge, or file collation, function is added. Input to the Merge is two or more ordered files; output is one ordered, merged file. The Merge can be invoked either as a command (merge) or as a subroutine (merge).

3. Support for the SORT portion of the ANSI COBOL Sort/Merge module, Level 2, is added.

4. Additional data types for keys and multiple key fields are supported. Release 3.0 supported only the character string data type and only a single key field.

5. Additional storage media and file organizations are supported for the input and output files. Any file can be supported which can be read or written sequentially via iox using any I/O module available at the Multics installation. The file is described by an attached description. Release 3.0 supported only sequential files in the Multics storage system, using vfile.
Additional Interfaces: The following user visible names are introduced.

1. Command
   merge

2. Subroutine Entry Points
   sort
   sort $release
   sort $return
   sort $initiate
   sort $commence
   sort $terminate
   merge
   (for COBOL only)

Control Arguments for Command:
-attach, -at
-sort_description, -sd
-merge_description, -md
-file_size
-merge_order
-string_size

Status Code and Message
The following message must be installed as a status code in the system error_table:

   data_seq_err   "Data sequence error."

DOCUMENTATION:

sort and merge commands, MPM Commands
sort and merge subroutine, MPM Subroutines
Sort/Merge PLM (planned)
6. The following additional user exit points are provided:

input_record exit: Permits the user to alter, delete, or insert records before they enter the sorting process. (This exit is not provided for the Merge.)

Output_record exit: Permits the user to alter, delete, insert, or summarize records coming out of the sorting or merging process before they are written to the output file.

7. Sequence checking of the output file is added.

8. The ability for the user to estimate the total amount of data to be sorted is added (file_size argument).

9. Command arguments for testing and performance measurement are added (-time, -merge_order, -string_size, -debug).

REASONS:

Required by Marketing for a "viable commercial system", and for certain contracts. See:


2. Multics Sort/Merge Project Summary and Authorization. PSA 224AD.

3. Product Calendar Item 2B0901A.

IMPLICATIONS:

Changed Interface: The keyword -sort_desc (-sd) must precede the pathname of the Sort Description when the Sort Description is supplied as a segment. In Release 3.0, the pathname of the Sort Description must be the first argument of the sort command and is not preceded by a keyword.
TITLE: FIX BUGS IN GCOS SIMULATOR

AUTHOR: R. H. Morrison

- Coded in PL/I
- Planned for System MR 3.1
- Fixes Bug Number(s) 75, 78, 82
- User/Operations-visible
- Performance: Better
- Replaces MCR

 Objections/Comments:

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

SUMMARY:

1. Change control tables to add implicit files and tab stop for column 73.
2. Change total restoration of machine conditions in gcos_mme_endc_ to partial restoration.
3. Recognize blank file codes.

REASON:

Fix bugs.

IMPLICATIONS:

Bugs will be fixed.

DETAILED PROPOSAL:

Replace:

```
gcos_control_tables_alm
```

```
gcos_mme_inos_.pl1
```

```
gcos_mme_fcon_.pl1
```

```
gcos_mme_fadd_.pl1
```
### edit_bugfile: list output @ 02/06/76 1452.2 mst Fri

<table>
<thead>
<tr>
<th>bug#</th>
<th>date</th>
<th>version</th>
<th>sts</th>
<th>log#/ /s</th>
<th>fixed/vers</th>
<th>resp</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcs00076</td>
<td>751105</td>
<td>3.0</td>
<td>x</td>
<td>/p 760204/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S. Linked GCOS jobs abort because no implicit H* file was allocated. Similarly, FILEDII aborts because of a* and s* files.</td>
<td></td>
</tr>
<tr>
<td>gcs00078</td>
<td>751110</td>
<td>3.0</td>
<td>x</td>
<td>/p 760204/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S. Tab settings for GCOS cards do not include &quot;73&quot; for the label field.</td>
<td></td>
</tr>
<tr>
<td>gcs00082</td>
<td>751112</td>
<td>3.0</td>
<td>x</td>
<td>/p 760204/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S. The tape reel sequence number check reports non-existent discrepancies to the terminal and requires a response (one of &quot;n&quot;, &quot;a&quot;, &quot;c&quot;, or &quot;i&quot;). R. Type &quot;i&quot;.</td>
<td></td>
</tr>
<tr>
<td>gcs00085</td>
<td>751119</td>
<td>3.0</td>
<td>x</td>
<td>/p 760204/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S. gcos_mme_endc restores all of the previous machine conditions saved from entering a courtesy call which causes Multics faults.</td>
<td></td>
</tr>
<tr>
<td>gcs00087</td>
<td>751120</td>
<td>3.0</td>
<td>x</td>
<td>/p 760204/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S. Blank file codes are not recognized as invalid.</td>
<td></td>
</tr>
</tbody>
</table>

**eb:**