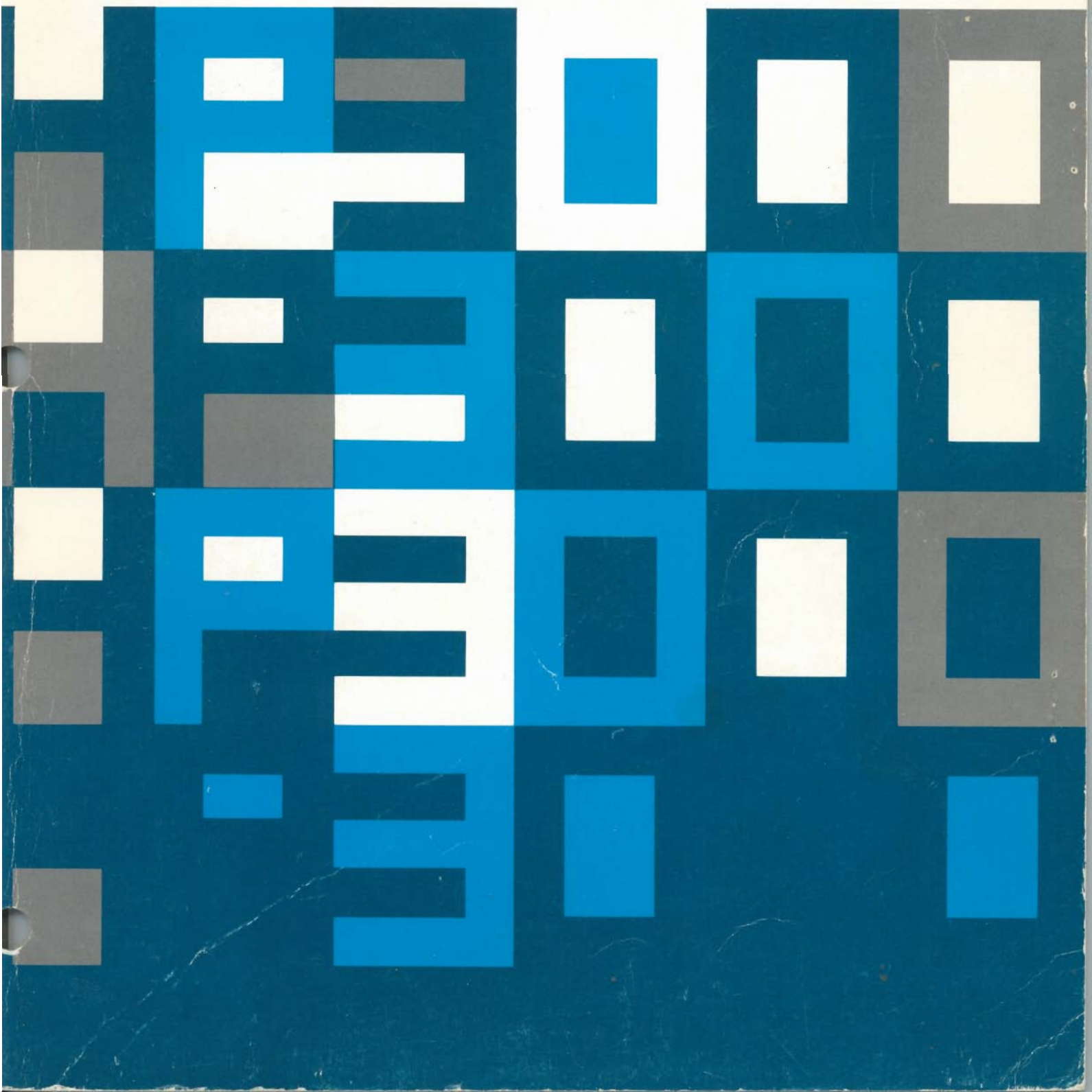
MPE III 2011

ROCKAWAY

HEWLETT PACKARD

COMMUNICATOR

ISSUE NUMBER 24



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

2011 INSTALLATION TAPE SYLLABUS - 22806B

May 1980

The 2011 Installation Tape contains many enhancements and corrective software changes. In order to help users sort out and learn about those that are most meaningful to each of them, the following syllabus is provided. This document summarizes changes, indicates types of users that would likely be affected, and indicates where more in depth information is to be found.

Existing Customer Training courses will be updated with the information pertaining to this IT. The nature of this release is such that a classroom course dedicated solely to 2011, such as the one offered for the 1918 IT, will not be necessary.

MPE

The following enhancements have been made in this release of the operating system.

ALL USERS
SYSTEM MANAGER
CONSOLE OPERATOR

- SHOWALLOW is a new command which allows users to determine which commands they have been ALLOWed by the console operator.

- CONSOLE is a new command which allows any user to determine on which terminal the console is residing.

REFERENCES: COMMUNICATOR #24, Page 7
MPE COMMANDS - 30000-90009
SERIES II/III CONSOLE OPERATORS GUIDE - 30000-900143
SERIES 30/33 CONSOLE OPERATORS GUIDE - 30070-90025

SYSTEM MANAGER
CONSOLE OPERATOR

- The SPOOLER has had many enhancements which make it more powerful and easy to use; the ALTSPOOLFILE command will now work with active spool files. STOPSPOOL will now stop printing active spool files immediately. OUTFENCE allows for setting individual outfences; SHOWOUT will display outfence values for each output device. OPENQ and SHUTQ options for STOPSPOOL have been reinstituted.

REFERENCES: COMMUNICATOR #24, Page 4
SERIES II/III CONSOLE OPERATORS GUIDE - 30000-90013
SERIES 30/33 CONSOLE OPERATORS GUIDE - 30070-90025

- SPOOK has also been enhanced to make it easier to use. It can now be run in batch mode as well as interactive mode. The ALTER, COPY AND APPEND commands now allow file specification by user, account, wild card or device file ID.

REFERENCES: COMMUNICATOR #24, Page 6
MPE III UTILITIES - 30000-90044

SYSTEM MANAGER
SYSTEM ANALYSTS
PROGRAMMERS

- Foreign Disc Facility (FDF) allows access to non-HP formatted discs and flexible discs. The foreign disc can be auto-recognized by MPE when mounted; it can also be accessed through the file system. Enhanced intrinsics and file options allow any type of programmatic manipulation of the foreign disc.

REFERENCES: COMMUNICATOR #24, Pages 8, 11
MPE COMMANDS - 30000-90009
MPE INTRINSICS - 30000-90010
SERIES II/III CONSOLE OPERATORS GUIDE - 30000-90013
SERIES 30/33 CONSOLE OPERATORS GUIDE - 30070-90025
SYSTEM MANAGER/SYSTEM SUPERVISOR - 30000-90014

PROGRAMMERS
SYSTEM MANAGER

- DPAN (Dump Analyzer) dialog and reports have been changed to make them easier to use.

REFERENCES: COMMUNICATOR #24, Page 9
MPE III UTILITIES - 30000-90044

- Information in tape labels is now read by FOPEN and can be reported by FFILEINFO.

REFERENCES: COMMUNICATOR #24, Page 6
MPE INTRINSICS - 30000-90010

SORT

REFERENCES: COMMUNICATOR #24, Pages 110, 160
SORT-MERGE Reference Manual, third edition
32214-90001

The new version of SORT-MERGE includes some major enhancements which are noted in the COMMUNICATOR, and documented extensively in the new edition of the SORT-MERGE Reference Manual. Several corrective software changes are also included in this version, and explained in the COMMUNICATOR.

OPERATIONS STAFF
SYSTEMS ANALYSTS
PROGRAMMERS

- SORT accepts multiple input files which are treated as one contiguous file; the characteristics of the first file specified are used for all files (input and output). This enhancement greatly simplifies sorting as only half as many input-output operations are required. See page 160 of the COMMUNICATOR for an interactive example of this.

SYSTEMS ANALYSTS
PROGRAMMERS

- ASCII input data can now be sorted or merged in ASCII or EBCDIC sequence, or a user-defined sequence. EBCDIC data can be sorted or merged in either EBCDIC or ASCII sequence.

PROGRAMMERS

- New MERGE intrinsics, similar to SORT intrinsics, have been added (MERGEINIT, MERGEOUTPUT, MERGEEND)

REFERENCES: COMMUNICATOR #24, Page 94
V/3000 Reference Manual, second edition, update #1
32209-90001

Five major enhancements and many corrective changes have taken place in V/3000. The enhancements are summarized below and described in detail in the Reference Manual. The COMMUNICATOR contains the NOON file which outlines all the changes.

SCREEN DESIGNERS
SYSTEMS ANALYSTS
PROGRAMMERS

- The MULTI-USAGE FORM FAMILY is an enhancement which is particularly useful for data base retrieval and update, and should improve performance of these applications. It is a collection of forms that have the same screen definition, but that may have different field specifications or form options.

  The first member of the family is called the PARENT FORM, the others are called SON FORMS. The SON FORMS essentially specify a new field specification set for the family.

  A son is reproduced from any member of the family by a son creation mechanism provided in FORMSPEC. The son acquires the field specifications of the member from which it was generated; the user can change the specs as desired. The screen definition of a SON cannot be changed. At run time, a SON FORM following any other member of the family, is equivalent to reusing the same screen definition, with a different set of field processing specifications. (See page 4-45a of Reference Manual.)

- Three new intrinsics have been provided to return information about the forms file on the file, form and field levels. The intrinsics have been structured so that the user determines exactly how much information will be retrieved on how many units (forms or fields). For more information on the VGETinfo intrinsics (VGETFILEINFO, VGETFORMINFO, VGETFIELDINFO) see page 6-25a of Reference Manual.

- An incremental compile capability has been implemented. When a forms file is compiled only the forms changed since the last compilation are compiled. The compilation step for the rest of the forms has been reduced to simple housekeeping functions. If a save field has been added, modified, or deleted, all forms are compiled. (See page 3-10 of the Reference Manual.)

PROGRAMMERS

- Forms files and reformat files are no longer KSAM files. All new files (slow and fast forms files and reformat files) created will be MPE files, but all existing KSAM files will still execute successfully. Also, a utility program will be provided to convert KSAM files to the new MPE format. (See page 3-9 of the Reference Manual.)

- A new intrinsic, VPOSTBATCH will improve the integrity of a batch file by posting an end of file mark after the last record referenced and updating the environmental information in the batch file user labels.

  The ENTRY program has been modified to call VPOSTBATCH after every 20 records. You may extend or shorten this posting interval by modifying the value of a globally declared integer. (See page 6-38a of the Reference Manual.)

****
** NOTE: Enhancement SR#9949, Item 7 on page 96 of the **
** COMMUNICATOR, is NOT implemented in this IT. **
** Please disregard Item 7. **
****

REFERENCES: COMMUNICATOR #24, Page 65
RPG Reference Manual 32014-90001 (update available Summer '80)

RPG PROGRAMMERS
SYSTEMS ANALYSTS

Two major enhancements have been made to RPG, as summarized below. Numerous minor enhancements and corrective changes have also been made. All are extensively documented in the NOON files which are printed in the COMMUNICATOR, beginning on page 65.

- More flexible mechanism for locking IMAGE data bases. Users now have the capability to:
  a. Lock entire IMAGE data base during the complete execution of a program
  b. Lock an IMAGE data set during complete execution of a program
  c. Lock a record whenever that record is accessed
  d. Lock the IMAGE data set whenever a record from that data set is accessed
  e. Enable locking so that user supplied routines (probably SPL) may perform all locking and unlocking

REFERENCE: COMMUNICATOR, Page 65

- Support for the Calculation operations TIME and TIME2 which return current time and date

REFERENCE: COMMUNICATOR, Page 75

****
** NOTE: Corrective Software Change SR#9929, Item 14 on page **
** 80 of the COMMUNICATOR, is NOT implemented in this **
** IT. Please disregard Item 14. **
****

APL

REFERENCE: COMMUNICATOR #24, Page 82

APL PROGRAMMMERS

Three significant changes have been made in APL. Numerous enhancements and corrective software changes have also been made. All are documented in the NOON files, printed in the COMMUNICATOR, beginning on page 82.

The major enhancements include:

- Function to provide a quick and easy way to send APL data to the line printer or to a disc file from an APL session or function (Page 82)
- Functions for character manipulations (Page 82)
- Performance improvements for membership, indexing and grading (Page 85)

ALEX ROCKAWAY

Table of Contents

Editor's Note

The note files and articles in this issue of COMMUNICATOR 3000 should give you a good idea of what Installation Tape 2011 is all about. There are several informative articles concerning the 2011 Release, and the note files are extensive, running from page 25 to page 139.

The opening article summarizes the collection of MPE III enhancements and gives examples of SPOOLER and SPOOK changes. Another MPE III enhancement, the Foreign Disc Facility, is discussed in depth in the article which begins on page 11.

The important SORT-MERGE/3000 enhancements are discussed and exemplified in an article beginning on page 161.

There are two articles concerning a new Hewlett-Packard product, COBOL II/3000. COBOL II/3000 is HP's high level implementation of the 1974 ANSI Standard. A general introduction to this powerful, new compiler begins on page 19, and a performance optimization discussion begins on page 158.

VIEW/3000's name has been changed to V/3000; this announcement is on page 10. The Documentation Section, beginning on page page 140, reports recent documentation activity. The latest Catalog of Customer Publications begins on page 147.

Editor
COMMUNICATOR 3000
HP General Systems Division
19447 Pruneridge Avenue
Cupertino, CA 95014

Many New MPE Enhancements Coming Your Way

By Pete Sinclair, Adrienne Bresso and Terry Ishida
General Systems Division

A number of significant new MPE enhancements are coming your way on Installation Tape (IT) 2011. Among the most significant of them are:

- Many new SPOOLER and SPOOK enhancements
- Tape label reading
- New Console locating and retrieval command
- New SHOWALLOW command
- New Association Table listing command
- Improved UDC logon performance
- Foreign Disc Facility
- Simplified DPAN dialogue

Each of these enhancements is described in more detail below. To get the complete story, though, we suggest that you read the documentation updates that you will receive with this IT and read the note files that appear in this issue. We're sure that you will be able to use many of these exciting new enhancements in your applications.

SPOOLER ENHANCEMENTS

There are many new enhancements to both the SPOOLER and SPOOK on the 2011 release of MPE.

- The ALTSPOOLFILE operator command **will now work** with active spool files. You no longer have to suspend spooling to alter the active spool file.
  Example:

```
:SHOWOUT SP
DEV/CL   DFID  JOBNUM  FNAME   STATE FRM SPACE RANK PRI #C
6        #O9   #S3     LP      ACTIVE       20    1   8  1
OUTFENCE = 1

:ALTSPOOLFILE #O9;COPIES=6     (alter active spoolfile)

:SHOWOUT SP
DEV/CL   DFID  JOBNUM  FNAME   STATE FRM SPACE RANK PRI #C
6        #O9   #S3     LP      ACTIVE       20    1   8  6
OUTFENCE = 1
```

- The **STOPSPOOL** operator command will now **stop printing** active spool files **immediately**, rather than waiting until they complete as it did in the past.
Example:

```
:SHOWOUT SP
DEV/CL   DFID  JOBNUM  FNAME   STATE FRM SPACE RANK PRI #C
6        #O9   #S3     LP      ACTIVE       20    1   8  6
OUTFENCE = 1

:STOPSPOOL 6

:SHOWOUT SP
DEV/CL   DFID  JOBNUM  FNAME   STATE FRM SPACE RANK PRI #C
6        #O9   #S3     LP      READY        20    1   8  6
OUTFENCE = 1
                    (the active spoolfile is stopped)
:SHOWDEV 6
LDEV  AVAIL    OWNERSHIP       VOLID          ASSOCIATION

  6   AVAIL                (the spooler is stopped)
```

- The **OUTFENCE** operator command has been **modified** to allow the operator to **set individual outfences for each output device as well as being able to set the fence globally** (as is currently done). The **SHOWOUT** command has also been **modified to allow it to display the outfence values for each output device.** This will allow you to control what priority is required to use each of the individual line printers on your system.
Example:

```
:OUTFENCE 14

:SHOWOUT SP
DEV/CL   DFID  JOBNUM  FNAME   STATE FRM SPACE RANK PRI #C
6        #O9   #S3     LP      READY        20    D   8  6
19       #O10  #S4     FASTLP  READY        20    D   8  1
OUTFENCE = 14

:OUTFENCE 6;LDEV=6     (lower fence for one printer)

:SHOWOUT SP
DEV/CL   DFID  JOBNUM  FNAME   STATE FRM SPACE RANK PRI #C
6        #O9   #S3     LP      ACTIVE       20    1   8  6
19       #O10   #S4    FASTLP  READY        20    D   8  1
OUTFENCE = 14
OUTFENCE = 6 FOR LDEV 6

:OUTFENCE 12           (reset all fences)
```

```
:SHOWOUT SP
DEV/CL    DFID   JOBNUM   FNAME    STATE FRM SPACE RANK PRI #C
6         #O9    #S3      LP       READY         20    1   8  5
19        #O10   #S4      FASTLP   READY         20    D   8  1
OUTFENCE = 12
```

- SPOOK can now be run in batch mode as well as interactively. You can now do your required SPOOK interactions programmatically, saving you the trouble of having to do them manually.

- The ALTER, COPY, and APPEND commands in SPOOK have been modified to allow files to be specified by either user, account, wild card (@), or device file ID's in addition to the current range and file name specifications. This change significantly enhances the flexibility of these commands.

- The OPENQ and SHUTQ options for the STOPSPOOL and STARTSPOOL operator commands have been reinstituted. The default for STOPSPOOL is SHUTQ while the default option for STARTSPOOL is OPENQ.
  Examples:

  ```
  :STOPSPOOL 6;OPENQ
   stops the printer  while still collecting spoolfiles for
   device 6.

  :STARTSPOOL 6;SHUTQ
   allows the spooler to be drained of spoolfiles while not
   collecting any new ones for DEV=6.
  ```

The net of these enhancements should be a more powerful and friendly SPOOLER and associated SPOOK utility, part of our continuing effort to make the MPE SPOOLER the best in the industry.

TAPE LABEL READING

The file system has been enhanced to use the information contained in the tape label (record size, record format, and block size) at FOPEN time in order to read the files on a labelled tape in the proper format. This is similar to the way disc file labels are interpreted when opening disc files. In addition, the new FFILEINFO intrinsic (introduced on the 1918 IT) has been enhanced to allow it to report key information contained in the tape label. These enhancements will make it much easier to keep track of and use tape label information progammatically.

NEW CONSOLE LOCATING AND RETRIEVAL COMMAND

By entering the new :CONSOLE command without specifying any parameters, MPE will respond by telling the user on which terminal device the console is currently residing. This new console locating capability is available to all system users (as compared to the :CONSOLE ldev operator command which is only available to certain users).
Example:

```
:CONSOLE
CONSOLE IS CURRENTLY ASSIGNED TO LDEV 20.
```

The :CONSOLE ldev operator command has been enhanced to allow the System Manager (SM) the ability to execute the command and obtain the console without having first been ALLOWed to use it. This added SM capability will make it easier for you to retrieve the console from downed terminal or modem lines as well as giving the System Manager ultimate control over where the console is.

NEW SHOWALLOW COMMAND

A new command called :SHOWALLOW has been added to allow users to determine which commands they have been ALLOWed to use by the console operator. Additionally, Account and System Managers will be able to determine what commands have been ALLOWed to their various users by specifying either the specific users or wild card (@) parameters when entering the :SHOWALLOW command. This new command will make it easier for you to keep track of which commands and to whom the operator commands have been distributed.
Example:

```
:SHOWALLOW

MANAGER.SYS
    USER HAS THE FOLLOWING COMMANDS ALLOWED:
STARTSPOOL     UP                ALLOW          ALTSPOOLFILE
BREAKJOB       DELETESPOOLFILE   STOPSPOOL      SUSPENDSPOOL
RESUMEJOB      RESUMESPOOL       CONSOLE        VMOUNT
THE FOLLOWING COMMANDS HAVE BEEN GLOBALLY ALLOWED:
UP             BREAKJOB          RESUMEJOB      VMOUNT
```

NEW ASSOCIATION TABLE LISTING COMMAND

A LIST entry point has been added to the ASOCTABL utility to allow users to obtain a formatted list of the data in the device association table (ASOCTABL). This will make it easier for the System Manager to determine which users may associate with which device classes.
Example:

```
:RUN ASOCTABL.PUB.SYS,LIST
LP=@.SYS
TAPE=@.SYS
TAPE=FIELD.SUPPORT
```

IMPROVED UDC LOGON PERFORMANCE

The algorithm that MPE uses to initialize UDC's when a user logs on has been modified to execute more efficiently. The speedup will be most noticeable to users which have a large number of UDC files set.

FOREIGN DISC FACILITY

The Foreign Disc Facility (FDF) is a set of MPE enhancements which permit non-privileged users to manipulate disc packs or flexible diskettes that do not have standard HP 3000 file system disc label formats.

A foreign disc is a new type of non-system domain disc similar to a serial disc or private volume. It can be created with VINIT, auto-recognized by MPE when mounted, and accessed through the file system. This allows any HP 3000 compatible disc pack or flexible diskette (i.e. readable by our drive electronics and on HP approved media) to be read or written to. The enhanced intrinsics and file options related to FDF can be used in your programs to manipulate the foreign disc files in any manner you desire.

You can use FDF for a number of applications. You can use it to read and write HP 250 or HP 300 diskettes on an HP 3000 Series 30 or 33. You can even use it to write your own custom private volumes facility on any of our disc drives with removable packs (7920 and 7925 on Series III; 7902, 7906, 7920, and 7925 on the Series 30/33). The options are limited only by the creativity of your programmers.

For further information on this exciting new enhancement, see the in-depth article on FDF in this issue of the COMMUNICATOR.

SIMPLIFIED DPAN DIALOGUE

The **DPAN (Dump Analyzer)** dialogue has been changed to make it easier to use. **All but one entry point** (EIGHTLPI - eight lines per inch) **and the PARM option have been deleted.** In addition, the DPAN reports have been enhanced to make them easier to read. Additional details can be found in the latest update for the DPAN section of the System Utilities Manual.

****

Announcing a New Name for HP VIEW/3000

by Jutta Kernke, General Systems Division

Due to a legal complication, we have decided to rename HP VIEW/3000 to:

HP3000 Data Entry and Forms
Management System (V/3000)

This new name will be used on all of our documentation, literature, procedures and policies as of March 1, 1980.

We will begin immediately to replace the name VIEW/3000 in our correspondence and daily usage to provide a smooth transition. Remember to start looking for V/3000 in your Software Status Bulletins and other support literature.

A Word About :DOWNLOAD

by Steve Zink, General Systems Division

The DEFAULT option in the :DOWNLOAD console operator command has been deleted because the VFC and the margin will not be reset between jobs as it was prior to the 2011 release. However, powering down the device, or reactivating the device RESET button, initializes the device environment to the default state.

Introducing the Foreign Disc Facility

A discussion of the usage of FDF -- a powerful, new facility which allows access to non-HP formatted discs.

by Adrienne Bresso, General Systems Division

The Foreign Disc Facility (FDF) is a set of MPE software enhancements which will permit non-privileged users to access and alter data residing on any removable disc (including flexible discs) even if the discs do not have standard MPE file system label formats.

For example, you will be able to read or write HP250, HP300 or even IBM3741 formatted diskettes on the Series 30/33 provided that you know the disc and diskette formats. The data is written to, or read from, a disc volume as though it were a standard MPE file. All that you need to do is write your own programs to interpret (if reading) or format (when writing) the data.

Before the 2011 release, it was not possible to access data on discs that had an unrecognizable label, i.e., a label that was not in a format known to MPE. MPE would only accept the following HP volume formats:

1. Valid private volume
2. Scratch volume
3. System volume
4. Unformatted volume
5. Serial disc volume

If a volume was mounted and was not in one of the above formats, you could either dismount the disc and try another one or write a new, recognizable label on it, thus destroying whatever information there might have been in track 0, sector 0. Either alternative defeated the purpose of accessing the data just as it was.

Discs that are not in one of the above formats are now referred to as "foreign" discs. FDF make it possible for MPE to automatically recognize foreign discs. The foreign disc is now considered another type of non-system domain disc, similar to a serial disc or a private volume.

Configuration Dialog

FDF provides for the designation of device classes as "foreign" classes in a manner similar to the way serial disc classes are defined. **A new query has been added to both SYSDUMP** (a routine for system backup) **and INITIAL** (an MPE procedure used at system cold load time). When adding a new device class name, the following I/O configuration dialog takes place:

```
        .
        .
ANY CHANGES? y
        .
        .
I/O CONFIGURATION CHANGES? y
LIST I/O DEVICES? y
LOG DRT U C T  SUB   TERM   REC   OUTPUT  MODE  DRIVER DEVICE
DEV  #  N H Y  TYPE         WIDTH   DEV          NAME CLASSES
 #      I A P
        T N E
        .
        .
3    57 0 0 2    0           128     0          IOFLOPO SDISC
                                                       FLOP
        .
        .
CLASS CHANGES? y
        .
        .
ADD CLASSES? y
CLASS NAME? FDISC
LOGICAL DEVICE #'S? 3
SERIAL DISC CLASS?  NO
FOREIGN DISC CLASS?  YES    <<<<This is the new query>>>>
        .
        .
LIST I/O DEVICES? YES

LOG DRT U  C T  SUB  TERM  REC   OUTPUT MODE DRIVER DEVICE
DEV  #  N  H Y  TYPE       WIDTH   DEV        NAME  CLASSES
 #      I  A P
        T  N E
        .
        .
3    57 0  0 2    0         128     0        IOFLOPO SDISC
                                                    FLOP
                                                    FDISC
        .
        .
ADD TO SYSTEM VOLUME SET?  NO
```

****

New VINIT Command >FOREIGN

Once a device class has been designated as a foreign class, it is then possible to mount any drive-compatible disc on the corresponding logical device and, by utilizing your own code, access and alter all information on that disc. To convert any disc format (i.e.,serial disc format) to a foreign disc, use VINIT (the volume initialization subsystem).

To create a foreign disc on logical device 3, you would mount a diskette on the drive and enter the following VINIT command:

```
>FOREIGN 3
```

Logical device 3 must be a non-system domain disc drive in the DOWN state. The action of this command is to fill the MPE label of a disc, which was either a previously formatted volume or not, with zeros, since all zeros is not a recognizable MPE label. (Any other nonrecognizable sector zero content would do as well.) This volume will then be recognized as a foreign disc with all subsequent usage. (The MPE label area is track 0, sector 0.)

The status of the foreign disc on logical device 3 can be displayed via the DSTAT command in VINIT. Here is an example of VINIT's DSTAT command showing the status of a foreign disc mounted on logical device 3 and still in a DOWN state:

```
>DSTAT 3
  LDEV-TYPE      STATUS      VOLUME (VOLUME SET-GEN)
 ___________    ________    _________________________
  3-7902         DOWNED      *FOREIGN DISC*
```

FDF does not include any new user commands. However, the user command "DSTAT" will display the "FOREIGN" status of a foreign disc, and has the same format as the VINIT "DSTAT" command.

New Console Command :FOREIGN

The mount/dismount procedures for foreign discs are almost identical to those for serial discs. Only the differences will be discussed in this article. When a foreign disc is mounted, the console will display this message:

```
FOREIGN DISC VOLUME ON LDEV# 3
```

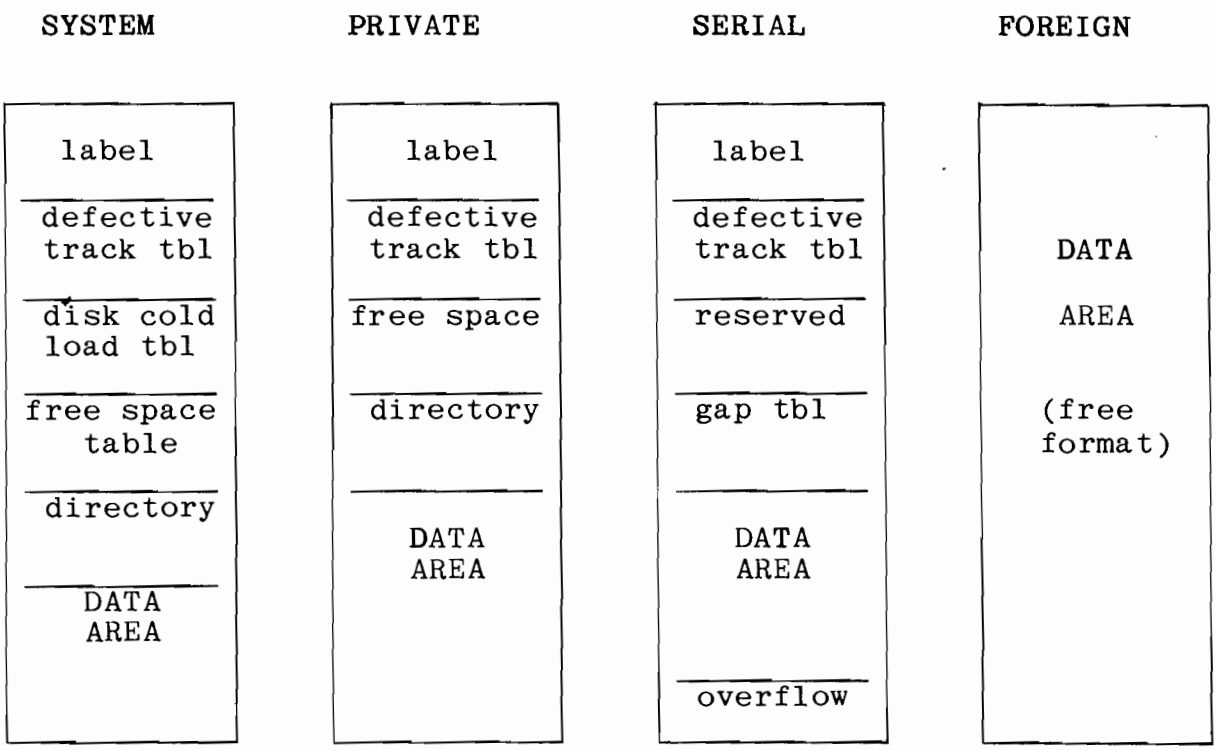
Since foreign discs are allowed to have completely arbitrary information formats, it is conceivable that a foreign disc could mimic a private volume, serial disc, etc. If a disc volume is erroneously auto-recognized, it will be necessary for the operator to intervene. To inform the system that a particular disc volume is in fact a foreign disc, a new operator command has been provided. The following example will show the transition from a serial disc to a foreign disc via the console operator command FOREIGN.

```
11:16/8/SERIAL DISC ON LDEV# 3
:DSTAT 3
   LDEV-TYPE          STATUS          VOLUME(VOLUME SET-GEN)
  -----------        --------        ------------------------
   3-7902             SERIAL           *UNALLOCATED*

:FOREIGN 3

:DSTAT 3
   LDEV-TYPE          STATUS          VOLUME(VOLUME SET-GEN)
  -----------        --------        ------------------------
   3-7902             FOREIGN          *UNALLOCATED*
```

The effect of this command is to force the system to treat the volume currently mounted on device 3 as a foreign disc. Logical device 3 must be in a foreign class, and the volume must be up but not in use.

```
********************* OPERATOR CAUTION ************************
*                                                             *
*   If the restrictions stated above are met, the system      *
*   will treat any volume mounted on the logical device as a  *
*   foreign disc.  As long as the volume is not a member of   *
*   the running system domain (i.e., not in the volume table),*
*   no security checking is done.  Therefore, any disc        *
*   (system, private volume, etc) could be accidentally       *
*   mounted and used as a foreign disc.  Since one            *
*   characteristic of FDF is the ability to write on the      *
*   MPE label area, file labels could be destroyed and        *
*   important information lost.                               *
*                                                             *
***************************************************************
```

COMPARISON OF FOUR MAJOR DISC FORMATS



As you can see, the foreign disc has a free format. What this means to you is that you can read and write at the beginning of the disc (track 0, sector 0). This illustrates one unique characteristic of Foreign Disc Facility.

Error Messages

There are seven error messages which may be produced by the FOREIGN command. Examples of these messages are as follows:

EXAMPLE ONE

```
:FOREIGN 3.3
          ^
FOREIGN EXPECTS EXACTLY ONE PARAMETER, THE DEVICE NUMBER
                                                (CIERR 3851)
```

EXAMPLE TWO

```
:FOREIGN 3
:DSTAT 3
  LDEV-TYPE        STATUS        VOLUME(VOLUME SET-GEN)
 -----------      --------      ------------------------
  3-7902           FOREIGN       *UNALLOCATED*

:FOREIGN 3
         ^
VOLUME IS ALREADY FOREIGN.  (CIWARN 3852)
```

EXAMPLE THREE

```
:FOREIGN 6        ***where device 6 is the line printer***
         ^
SPECIFIED DEVICE MUST BE A MOVING HEAD DISC. (CIERR 3850)
```

EXAMPLE FOUR

```
13:01/8/DISMOUNT ON LDEV# 3

:FOREIGN 3
         ^
NO VOLUME IS MOUNTED ON DRIVE   (CIERR 3853)
:FOREIGN 1
         ^
SPECIFIED DEVICE IS IN THE SYSTEM DOMAIN.  (CIERR 3856)
```

The remaining two messages are self-explanatory:

```
     VOLUME IS AN IN-USE PRIVATE VOLUME
     VOLUME IS AN IN-USE SERIAL DISC
```

FDF and File System Intrinsics

When accessing and altering foreign disc files programmatically, it is necessary to use the file system intrinsics; therefore, they are discussed below as they are affected by FDF.

FOPEN:

To FOPEN a foreign disc, the DEVICE parameter must specify either a foreign disc class name or the "ldn" of a disc in a foreign disc class. The device parameter may also be overridden by a

file equation specifying such a class name or logical device number.

When the FOPEN is executed, the operator will be interrogated as described above. If a foreign disc is to be FOPENed by logical device number, it is necessary for the volume to be mounted BEFORE the FOPEN takes place. Otherwise, FOPEN will assume a serial disc is to be mounted by default. This problem may be avoided by opening the disc by class name.

The FOPEN parameters affected by FDF are as follows:

FOPTIONS	Fixed, binary records without carriage control are forced.
AOPTIONS	All access types are allowed except for append and execute.
RECSIZE	Forced to 128 words. IBM diskettes are forced to 64 words.
DEVICE	Must either be a foreign disc class name or the logical device number of a disc in a foreign disc class.
BLOCKFACTOR	>= 1.
FILESIZE	Set to the maximum physical size of the disc as determined by its subtype; but because of spare tracks, etc., the logical size will usually be less than the physical size.

The other parameters are ignored by FDF.

FCLOSE:

There are no changes to FCLOSE except that FDF ignores the DISPOSITION and SECURITYCODE parameters.

FREAD, FWRITE, FREADDIR, FWRITEDIR, FUPDATE:

These intrinsics work in the usual manner. Record 0 corresponds to sector 0, track 0, cylinder 0, and subsequent records correspond to subsequent sectors, tracks, and cylinders in that order. The last sector of the last track of the last cylinder is the last record of the file, and any attempts to read or write beyond this point will result in an end-of-file.

Handling IBM diskettes is a little different. IBM diskettes number sectors starting with one rather than zero and the diskette driver increments all sector addresses by one. Therefore, record number zero would be specified to read sector one on an IBM diskette, etc.

FGETINFO:

A new device type, %07, is returned by FGETINFO for foreign discs. This is similar to the use of device type %37 returned for serial discs.

The following FGETINFO parameters are not meaningful for a foreign disc:

FILE CODE	USERLABELS
EOF	CREATORID
EXTSIZE	LABADDR
NUMEXTENT	

FCHECK:

Defective tracks processing is not done by the system on foreign discs. An error code is returned by FCHECK when an error occurs which would normally result in a defective track table entry.

FFILEINFO:

There are three new codes returned by FFILEINFO:

40 - returns disc or diskette device status as a double word. For non-disc and non-diskette devices 0D is returned.

41 - returns actual device type, as opposed to class access type, as an integer.

42 - returns device subtype as an integer.

****

FDF's ability to communicate with media in formats other than standard HP formats provides greater flexibility and efficiency for data transfer and interchangeability. With the introduction of FDF, the user now has the opportunity to alter and access data in any format that is known to him.

A Look at COBOL II/3000

An overview of capabilities, language extensions, system integration, conversion and environment.

by John Santeler, General Systems Division

COBOL II/3000 is the primary commercial language for the HP3000 computer system. As the compiler most frequently used in the development of business applications, COBOL II is implemented at the highest level for 9 of 12 functional processing modules of the 1974 ANSI Standard for COBOL.

In addition, COBOL II has extended capabilities beyond the ANSI 74 requirements. As a subsystem on the HP3000, COBOL II interfaces with the data communication and data base management subsystems, as well as with the MPE file system. These extensions reduce programming effort and application development time by providing a complete and comprehensive set of on-line application development tools for the COBOL programmer.

Implementation Level

COBOL II/3000 conforms to the Level-2 implementation (except the RERUN option for I/O) of nine of the twelve modules defined by the ANSI COBOL X3.23-1974 specifications. This Level 2 implementation of each module provides the programmer with the widest range of processing capabilities and options as defined by the Standard. An overview of these capabilities provided to the programmer follows:

Nucleus Module:	Defines structures for elementary internal processing of data within the basic structure of the four divisions of a COBOL program.
Table Handling Module:	Provides the capability for defining and accessing items in up to three dimensional, variable-length tables.
Sequential I/O Module:	Provides the capability for defining and accessing sequentially organized files.

Relative I/O Module:	Provides the capability for defining and accessing files in which records are identified by relative record number.
Indexed I/O Module:	Provides the capability of defining and accessing files in which records are identified by key values as well as sequentially and randomly within the same COBOL program.
SORT-MERGE Module:	Provides sorting and merging capabilities of single and multiple input files. ASCII and EBCDIC data types and user defined collating sequences are implemented to allow the tailoring of the SORT-MERGE capability to specific data and sequence requirements of the particular application.
Segmentation Module:	Allows for specification of the segmentation of the COBOL program.
Library Module:	Reduces redundant code within multiple programs by allowing the programmer to place frequently accessed code into one or more library files which are incorporated into the program during compilation.
Inter-Program Communication Module:	Allows COBOL to transfer control to one or more programs which may be written in COBOL (either COBOL/3000 or COBOL II/3000), SPL or FORTRAN.

Language Extensions

In addition to meeting the Level-two ANSI Standard for nine functional processing modules, Hewlett-Packard has implemented several extensions to the Standard in order to increase the capability and efficiency of COBOL II/3000 application. The most significant extensions are:

Microcoded Instructions: Frequently utilized COBOL instructions are coded into system firmware to improve the RUN-TIME efficiency of application programs.

* The microcoded PERFORM statement decreases the overhead for transferring control to other sections of code within structured programs.

* The EDIT instruction which produces formatted output is microcoded to increase the efficiency in producing output reports.

Pre-Processor Function:

Provides statements which allow the programmer to equate a particular section of code ($DEFINE) or a file ($INCLUDE) to an identifier. This identifier can be referenced throughout the program and at compilation is replaced with the expanded code or program file which it represents.

Program Debugging Aids:

Allows for the increased efficiency of the COBOL programming staff by providing tools which aid in finding problems within a program during the development cycle.

* CROSSREF provides a listing of all symbols and labels and their locations within the source code.

* VERBMAP provides a listing of all COBOL II statements and their location within the source code.

* DEBUG allows a programmer to interactively trace through the execution of a program in order to determine logic errors.

Access to Subprograms:

COBOL II provides the capability to call subprograms written in COBOL, COBOL II, FORTRAN and SPL. This allows for the usage of pre-existing code which performs a desired function.

Access to MPE System Intrinsics:

Allows easy programmatic access to the full power of the MPE operating system through direct access to the System Intrinsics.

* Allows for passing byte and word address parameters and usage of decimal or octal literals.

* Allows checking of Condition Codes indicating the result of Calling the Intrinsic.

ACCEPT FREE option:	Allows a free format for low-volume data entry.
File Locking Capability:	Permits a program to request exclusive access to an MPE or KSAM file through the usage of the EXCLUSIVE and UNEXCLUSIVE statements.
Special Registers:	The frequently used registers TALLY, CURRENT-DATE, TIME-OF-DAY, and WHEN-COMPILED are provided by COBOL II to reduce programming effort by eliminating the need to construct and access these registers within the WORKING SECTION of the program.
Packed Decimal (COMP-3):	Allows for efficient storage of data on the stack during execution.
Multiple Entry Points Subprograms:	Reduces code redundancy by allowing similar functions (within portions of code which are identical) to be coded into a single subprogram and be accessed through multiple entry points.

SYSTEM INTEGRATION

COBOL is the primary language and program development tool for designing applications in an on-line business environment. The programming capabilities provided by the ANSI Standard and Hewlett-Packard's extensions to the Standard give the programmer the basic elements for data manipulation. Additionally, COBOL II provides an interface to all of the data management and data communication tools available on the HP3000 computers.

For data management COBOL II provides:

*Access to both sequential MPE and indexed (KSAM) files through the usage of ANSI Standard COBOL Input and Output operations.

*Access to V/3000, Hewlett-Packard's data entry product, as well as IMAGE/3000, Hewlett-Packard's data base management package through usage of procedure libraries.

CONVERSION

Conversion from COBOL/3000, Hewlett-Packard's implementation of the ANSI 68 COBOL Standard to COBOL II/3000 can be achieved by usage of the Conversion Guide (32233-90005) and the program COBCNV, which is standard with purchase of COBOL II/3000. COBCNV flags the syntax changes in COBOL/3000 programs which do not conform to the 74 Standard. (Primarily new reserve words.) These programs may then be modified, recompiled and executed using COBOL II.

HARDWARE ENVIRONMENT

The minimum hardware system required to implement COBOL II/3000 is an HP3000 Series II, III, 30, or 33 with 256KB of memory. However, the amount of memory actually needed is dependent upon the overall system load and performance expectations which should be determined in consultation with the local sales office.

SOFTWARE ENVIRONMENT

COBOL II/3000 requires the current version of the MPE III operating system. KSAM/3000 is required for compilation in the fundamental operating system. This product is included for all systems shipped as of August 1, 1979.

DOCUMENTATION

The documentation provided with each order of the COBOL II/3000 product (32233A) is:

COBOL II/3000 Reference Manual (32233-90001)
COBOL/3000 to COBOL II/3000 Conversion Guide (32233-90005)

ORDERING INFORMATION

COBOL II/3000 is available according to the terms & conditions of the Hewlett-Packard Software Purchase Agreement. The agreement defines the initial payment and monthly fees for the standard product.

Product Number	Purchase Description
32233A	COBOL II/3000 Right-to-Use
32233R	COBOL II/3000 Right-to-Copy
32233T	CSS for COBOL II/3000
32233V	CSS for an additional system: COBOL II/3000
32233S	SSS for COBOL II/3000
32233Q	MUS for COBOL II/3000

In addition to the above products, systems shipped prior to January 1, 1980 require a firmware instruction set:

Product Number	Purchase Description
32234A	COBOL II/3000 Extended Instruction Set (Series II/III)
32236A	COBOL II/3000 Extended Instruction Set (Series 30/33)

For customers converting from COBOL/3000 an option for CSS is available to provide simultaneous support for both products.

32233T option 68	CSS for COBOL/3000 simultaneous with with COBOL II/3000
32233V option 68	CSS (central site) for COBOL/3000 simultaneous with COBOL II/3000

MPE III Series II-III Software Update

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES II/III
****

CONTENTS OF INSTALLATION TAPE DATE CODE '2011'

PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED BY THIS M.I.T. AND ALSO REFERENCE PERTINENT NOTE FILES CONTAINING INFORMATION ABOUT THE MODIFICATIONS. THESE FILES MAY BE LISTED USING EDITOR OR FCOPY.

****

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*MPE	32002B	01.01	2011
*SEGMENTER	32050A	01.02	2011
*SPL	32100A	08.00	2011
*BASIC	32101B	00.12	2011
*FORTRAN	32102B	01.03	2011
*BASIC COMPILER	32103B	00.12	2011
*RPG	32104A	04.05	2011
*APL/3000	32105A	01.03	2011
BUILDINT	32150A	03.01	1623
*DS/3000	32190A	02.06	2011
MRJE	32192A	01.00	DATACOMM
*MTS	32193A	01.03	2011
*EDITOR	32201A	07.07	2011
SCIENTIFIC LIBRARY	32205B	00.04	1906
*DEL/3000	32206A	01.10	2011
*KSAM/3000	32208A	03.01	2011
*V/3000	32209A	01.01	2011
*COMPILER LIBRARY	32211D	00.10	2011
*FCOPY	32212A	03.11	2011
*COBOL	32213C	02.05	2011
*SORT/MERGE	32214B	02.03	2011
*IMAGE	32215B	02.05	2011
*QUERY	32216A	04.03	2011
TRACE	32222A	03.03	1814
*XA2100	32223A	01.04	2011
XL2100	32226A	02.00	1636
*COBOL LIBRARY	32232A	00.01	2011
*COBOLII	32233A	00.01	2011
PROG CONTROLLER 30300B/30361B-BCS	30361B	00.00	1621
PROG CONTROLLER 30301B/30361B-1-RTE	30361B-1	00.02	1701
RJE 2780/3780	30130E	00.03	DATACOMM
CALCOMP PLOTTER	30126A	00.01	1640
*DIAGNOSTICS	32230A	-- --	2011

DIAGNOSTIC INFORMATION IS CONTAINED IN THE FILE NOON230A.

* NOTE FILES(NOONYYYZ) CONTAIN THE CHANGE INFORMATION

WHERE YYY =LAST THREE DIGITS OF THE PRODUCT NUMBER.
(E.G. MPE IS HP32002. THEREFORE YYY=002.)
Z =CURRENTLY RELEASED VERSION DIGIT OF PRODUCT.

MPE HP32002B.01.01
NOONO02B.HP32002.SUPPORT

I. MPE 32002B.01.01

A. MODULES MODIFIED B.01.01

MODULE		CHANGE HISTORY												
NAME	NO	A.01.XX			B.00.XX			B.01.XX						
		1	2	MR	0	1	2	0	1	2	3	4	5	6
INITIAL	0	X		X	X	X	X	X	X					
SYSDUMP	1	X		X	X	X	X	X	X					
SEGPROC	2								X					
SEGDVR	3								X					
DISPATCH	4				X			X	X					
LOAD	5				X	X	X	X	X					
UCOP	7				X	X	X		X					
DEVREC	8				X	X	X	X	X					
PROGEN	9	X		X	X	X	X	X	X					
ININ	10	X	X		X		X	X	X					
MEMLOGP	11				X			X						
LOG	12		X		X				X					
IOPTRDO	13				X									
IOPTPNO	14		X		X		X							
IOPLOTO	15				X									
IOMDISCO	16													
IOFDISCO	17													
IOTAPEO	18	X	X		X		X	X	X					
IOLPRTO	19				X		X	X	X					
IOCDRDO	20				X		X	X	X					
IOTERMO	22	X	X		X		X	X	X					
IOPRPNO	24		X		X		X	X						
IOREMO	25													
IOMDISC1	27		X		X				X					
PFAIL	30						X							
PVPROC	31				X	X			X					

MODULE		CHANGE HISTORY												
NAME	NO	A.01.XX			B.00.XX			B.01.XX						
		1	2	MR	0	1	2	0	1	2	3	4	5	6
VINIT	32				X	X	X		X					
MAKECAT	40				X		X	X	X					
FILESYS	50	X	X	X	X	X	X	X	X					
COMM'INT	51	X	X	X	X	X	X	X	X					
STORE/RESTORE	52	X			X		X	X	X					
DIRC	53				X	X	X	X	X					
ALLOCATE	54	X	X		X	X	X	X	X					
HARDRES	55								X					
SOFTRES	56	X	X		X	X	X	X	X					
MMDISKR	57	X	X		X	X	X	X	X					
ABORTDUMP	58		X		X		X		X					
MESSAGE	59				X	X		X	X					
PROCSEG	60	X	X		X		X	X	X					
NRIO	62	X	X		X		X	X	X					
PCREATE	63		X		X		X		X					
MORGUE	64	X	X		X		X	X	X					
DATASEG	67	X	X		X	X	X	X	X					
CHECKER	69				X	X	X		X					
UTILITY	70	X	X		X			X	X					
SEGUTIL	71								X					
LOADER1	72		X		X	X	X	X	X					
RINS	73					X	X	X	X					
JOBTABLE	74	X	X		X	X	X	X	X					
DEBUG	75				X		X	X	X					
NURSERY	76				X	X	X	X	X					
FIRMWARESIM	78				X									
SPOOLING	79	X	X		X		X	X	X					
SPOOLCOMS	80	X	X		X		X	X	X					
PVSYS	81				X	X	X		X					
UDC	82				X		X	X	X					
USER	83				X				X					
HELPUSER	84				X				X					
OPCOMMAND	85							X	X					
LABSEG	86				X	X	X	X	X					
SDISC	87				X	X	X	X	X					
MEASIO	88							X	X					
LOGSEG0	90							X	X					
LOGSEG1	91							X	X					
CATALOG					X	X	X	X	X					
CICAT					X		X	X	X					

SYSTEM	LAST CHANGE NUMBER
B.00.00	0066
B.00.01	0134
B.00.02	0472
B.01.00	0789
B.01.01	1261

NOTE: Each change made to MPE is now identified by a unique change number in columns 64/72 (eg <<00120>>). This matrix provides the range of the change numbers used to build each version of MPE.

B. ENHANCEMENTS

FIX NUMBER — DESCRIPTION

790. INITIAL (00)
Added device type 17 processing code for INP.
Expanded CSLDTX table for new DATACOMM console command -- SHOWCOM.
Deleted requirement for loading all driver code into BANK 0.

790. SYSDUMP (01)
Added device type 17 processing code for INP.

790. PROGEN (09)
Added new DATACOMM console command -- SHOWCOM.

790. FILESYS (50)
Added IOWAIT changes for INP.

819. MMDISKR (57)
ADDTOWS was modified to execute with interrupts enabled. This reduces the length of time that interrupts are kept disabled.

819. DISPATCH (04)
DISPATCH now excutes for a short period of time with interrupts disabled.

FIX NUMBER	DESCRIPTION
819.	SOFTRES (56) ADDTOWS was modified to execute with interrupts enabled. This reduces the length of time that interrupts are kept disabled.
828.	FILESYS (50) Implemented tape label information in new FILE/INFO intrinsic.
834.	CI (51) An appropriate error message is now given for attempts to create circular :FILE and :CLINE equations.
841.	FILESYS (50) Added a feature to have blocksize, record size, and record format in labeled tape override the information in :FILE EQUATION and "FOPEN" in the same hierarchy applied to disc files.
857.	SOFTRES (56) A process in the "DS" subqueue now will be given TPRI priority whenever it does a terminal read.
858.	SPOOLCOMS (80) Allowed MRJE to change user and account of output spoolfile from "manager.sys".
874.	OPCOMMAND (85) It is now possible to specify the OUTFENCE for each ldev individually.
874.	CATALOG OUTFENCE was changed so that each ldev can be set individually.
874.	SPOOLING (79) OUTFENCE was changed so that each ldev can be set individually.
874.	SPOOLCOMS (80) OUTFENCE was changed so that each ldev can be set individually.
893.	OPCOMMAND (85) The CONSOLE command with no parameters now reports where the console is.

FIX NUMBER	DESCRIPTION
894.	CATALOG Added the :SHOWALLOW command.
896.	SPOOLING (79) The operator is now informed of an I/O error during a power fail or VFC reset (2608).
896.	CATALOG The operator is now informed of an I/O error during a power fail or VFC reset (2608).
925.	INITIAL (00) INITIAL will now print the following message CORE RESIDENT MPE =XXXX BANK 0 USED DURING BOOT=XXXX when loading from tape or making any changes. This is intended to be an aid for MIT installation.
1025.	INITIAL (00) Deleted unused equates and defines so that a stack overflow will not occur during the crossreference listing in the job file.
1036.	INITIAL (00) Prevented disc free space table overflow during INITIAL. If the table is full, additional space is lost.
1038.	SYSDUMP (01) Provided a date stamp for the list of files dumped.
1042.	OPCOMMAND (85) 1. Allowed anyone to find out where the console is. 2. Allowed anyone with SM capability to use the :CONSOLE command.
1043.	COMMAND INTERPRETER (51) Allowed anyone to find out where the console is.
1044.	CATALOG Added messages for the LIST entry point in ASOCTABL.
1076.	CATALOG These additional error messages were added: FILE SYSTEM: Number of opens for file exceeds 255 (FSERR 300) PRIVATE VOLUMES: Private Volume user Segment (Internal Problem) Full (PVERR 54).

FIX NUMBER	DESCRIPTION
1088.	OPCOMMAND (85) The following spool commands were added: :STARTSPOOL 6 [;SHUTQ] :STOPSPOOL 6 [;OPENQ]
1089.	CATALOG Error messages were added relating to the SHUTQ/ OPENQ parameters of :STARTSPOOL and :STOPSPOOL.
1106.	PVSYS (81) The Private Volume user table was made expandable (DST 66) in increments of 128 words up to 4k.
1112.	NURSERY (76) During job/session initiation, the error messages which report to the user and the console that the account/group is out of time have been expanded to differentiate between CPU and CONNECT time.
1147.	HARDRES (55) TICK is changed so that a user can programmatically alter the system clock interrupt rate. This will eliminate the extra clock for the same purpose.
1165.	COMMAND INTERPRETER (51) 1. Added CI routines for IML. 2. Fixed CLINE misc. array bug.
1175.	FILESYS (50) FOPEN no longer requires that "filecode" be passed to open a priviledged file. If not present, the FOPEN will succeed as long as the caller is privilieged. If filecode is supplied and the file is priviledged, the supplied filecode must match the filecode in the file's label.
1176.	PROGEN (09) CTRL-A command executor will no longer recognize =DSLINE as legal and will print out an informatory message to use :DSCONTROL.
1177.	CI (51) Command lookup in COMSEARCH will now recognize :DSCONTROL as a legal command and will call CONSDSLINE'.
1178.	CATALOG Messages added for :DSCONTROL. Some existing messages were rewritten for clarity.

FIX NUMBER	DESCRIPTION

1200. LOADER1 (72)
Changed so that the stack of a new process is built with at least 2 additional words between DB and the Q-initial region. These additional words are placed after the global DB region of the stack and before the PARM word at Q-4. Thus, the DB global data lay-out remains the same; however, the location of Q-initial is at least 2 words greater than before.

NOTE: Programs which assume that the DB to the Q-initial region of the stack remains fixed in size, and use knowledge of that size to generate Q+ addressing, will no longer produce the correct addresses. The only valid way to produce DB relative Q+ addresses is with the LRA instruction - not with hard coded constants.

1206. CATALOG
Added error messages for the new MPLINE command.

1207. PROGEN (09)
Allowed MPLINE command to be accessed as a normal command.

1208. CI (51)
Allowed MPLINE command to be accessed as a normal command.

1224. SOFTRES (56)
Performance enhancement for EXCHANGDB. The ADDTOWS call in EXCHANGDB was replaced by a call from MAM on the extra data segment PREP, and by a call from DISPATCH on process quiesce. MMSTAT was optimized for normal (positive) event logging.

1233. NRIO (62)
Terminal type 6 will be allowed to run at speeds of 10,15,30,60,120, & 240 char/sec.

1234. NRIO (62)
Allowed for creation of new terminal type 18.

1237. PROGEN (09)
Changed the command SHOWCOM from PROGEN to the Command Interpreter.

FIX NUMBER	DESCRIPTION

1238. CI (51)
Allows users to access SHOWCOM from the Command Interpreter.

1239. OPCOMMAND (85)
Allows users to use the SHOWCOM command from the Command Interpreter.

1242. HARDRES (55)
Creation of a new terminal type 18 which will run at speeds of 10,15,30,60,120,& 240 char/sec. Terminal type 18 will not send ENQ's and will not send a DC1 to start reads. Terminal type 18 will send no syncs after returns or line feeds.

1246. LOADER1 (72)
Initialized variable in the procedure LOAD for correct computation of stack space required to load new process code.

C. CORRECTIVE SOFTWARE CHANGES

800. CATALOG
Corrected msg 12, set 1 to show card column number of invalid hollerith character. (SMR # 5939)

802. NURSERY (76)
The group password was not required when the user loged on to his home group. If the home group is explicitly given in the :HELLO or :JOB command, a passwored is required. (SMR # 6798)

803. DEBUG (75)
Fixed release of console mode when debugging from a system process.

804. RINS (73)
A new intrinsic LOCRINOWNER now has error checking facilities to prevent abnormal termination of rolling process and possible system failure.

805. FILESYS (50)
When logged on to a PV group, :BUILD did not always succeed when DEV=DISC was specified. An erroneous message, out of DISC SPACE was returned. (SMR # 5907) This has been corrected.

FIX NUMBER	DESCRIPTION
806.	FILESYS (50) Creator and account names are now both checked for a match before allowing a RENAME. (SMR # 4845)
807.	LOAD (05) An extra data segment cache has been introduced to hold SL.PUB.SYS directory entries.
808.	LABSEG (86) This fix corrects a positioning problem which occurred when no user labels written on an IBM standard labeled tape. (SMR # 6527)
810.	USER (83) The user intrinsic, GENMESSAGE, has been corrected so that when the message length is greater than the supplied buffer length, the message is buffered up to the size of the buffer and the buffer size is returned as the length. (SMR # 6732)
812.	MAKECAT (40) MAKECAT now runs with traps turned off to prevent system failure 311.
813.	MESSAGE (59) 1. It is now possible to =ABORTJOB on jobs which have REPLY requests pending. 2. =LOGOFF and =SHUTDOWN no longer hang the system when REPLY requests are pending. (SMR # 6445)
813.	MORGUE (64) See the description for this change in fix #813 listed above. (SMR # 6445)
814.	STORE/RESTORE (52) An arithmetic error was corrected in STORE so that the last extent size is converted to logical before being converted to double, so that it is no longer made negative when >32k.
820.	MESSAGE (59) When using either the "DIR" or "BUILD" entry of MAKECAT.PUB.SYS., the new catalog will not be purgeable (the old one will be). (SMR # 6898)
824.	LOADER1 (72) The loader wil no longer accept a non-disc file as a program file.
830.	COMMAND INTERPRETER (51) When break follows a PREP or PREPRUN command, REDO will now show the previous command as "PREP".

FIX NUMBER	DESCRIPTION
832.	COMMAND INTERPRETER (51) The condition code is now checked after the READs in :REDO and :PURGEUSER. This prevents infinite looping in the event of I/O errors or EOF's (BYE, HELLO,JOB,DATA,:EOF:). (SMR #6323)
833.	COMMAND INTERPRETER (51) The :ALLOCATE/:DEALLOCATE commands now give CIERR error messages if an error occurs. (SR # 6694)
835.	COMMAND INTERPRETER (51) IF's and ENDIF's must now be properly paired within User Defined Commands and while in break mode. UDC's that are in the FALSE part of :IF statements are no longer expanded.
835.	UDC (82) See CI description, fix #835.
837.	VINIT (32) VINIT will now fix PFSPACE to report correctly when the free space table has 0 entries because the disk is completely full. (SMR # 6552)
839.	SOFTRES (56) System failure #108 could result from improper memory link address calulation (tested as an INTEGER instead of LOGICAL). This fix corrects this problem. (SMR # 7204)
840.	ABORTDUMP (58) Supplying an invalid (external) plabel in the XARITRAP intrinsic for enabling user traps could cause system failure 311 when the process handled the trap. This fix causes the process to be aborted with "invalid STT".(SMR # 7410)
843.	SOFTRES (56) MAM lock requests will no longer allow non-lock requests to run concurrently. (SMR #'s 6134, 6135,6136,6137)
844.	SYSDUMP (1) When adding a disc to the system and the device class name was greater than 8 characters, the question "IS classname A SERIAL DISC CLASS?" would print out garbage for the classname. This has been fixed. (SMR # 6768)

FIX NUMBER	DESCRIPTION
845.	SPL INTRINSIC FILE The PUTJCW SPL intrinsic now expects the second parameter, JCWVALUE, to be type logical.(SMR # 7117)
846.	ALLOCATE (54) Prevents SF353.
847.	IOTAPE0 (18) Added an extra test for diagnosing intermittent hardware faults.(7017)
849.	COMMAND INTERPRETER (51) Messages have been added to the CI which will indicate the flow of command execution for the :IF and :ELSE commands. For example, after the :IF, command a message will be given indicating whether the expression was true or false. (SMR # 6555)
850.	COMMAND INTERPRETER (51) When a user logs onto an APL terminal using the (APL) facility, the welcome message will now be translated. (SR # 6862)
851.	COMMAND INTERPRETER (51) In a UDC with option NOBREAK, the NOBREAK now remains in effect even if a program within the UDC enables a break. (SR # 6704)
852.	COMMAND INTERPRETER (51) LISTF now opens the listfile as a new file. When the last file is closed, it is now saved in the temporary file domain. (SR # 5841)
854.	COMMAND INTERPRETER (51) The :QUANTUM command no longer is modified if an error occurs (for example, REDO will echo what is typed).
855.	COMMAND INTERPRETER (51) SF505 no longer occurs when doing an ALTGROUP; VS=VSET: ACT for a group in the system domain which is not empty. (SR # 6703)
859.	RINS (73) FREERIN was fixed to ensure that the specified RIN is within the configured number of RINS. (SMR # 7580)
863.	UDC (82) An option logon UDC at the account (or system) level will no longer cause a UDC by the same name at the user (or account) level to be executed at logon. (SMR # 7648)

FIX NUMBER	DESCRIPTION
866.	VINIT (32) Allows the PVINIT DTRACK function to work correctly when several tracks are reassigned and a large number of files are purged.
867.	ABORTDUMP (58) The intrinsic STACKDUMP no longer truncates digits from the ID number in the title output.
868.	ALLOCATE (54) Fixed ALLOCATE so that when atempting to ALLOCATE a real device, IDD entries do not refer to spoolfiles. Also fixed ALLOCATE so that when attempting to ALLOCATE a virtual device, IDD entries do not refer to real devicefiles.
869.	SYSDUMP (01) Sysdump has been preped with MAXDATA=24000 instead of MAXDATA=16000 to allow for larger I/O configurations. (SMR # 7876)
872.	USER (83) An integer overflow condition in the intrinsic GENMESSAGE has been corrected.(SMR # 7885)
877.	IOCDRDO (20) This change put the KATAKANA changes back into the card reader driver. It allowed data cards encoded with IBM's KATAKANA-HOLLERITH (IBM would convert this code to EBCDIC) to be converted to JIS code (MPE compatable), if subtype=1. (SMR # 7476)
879.	CI (51) For :NEWUSER/:NEWGROUP, if you specify a capability which does not belong to the account, the USER/GROUP only receives capabilities which were both specified and belong to the account.(SMR # 7544)
884.	CATALOG New CIWARN 1904 will warn the user that UDC'S are no longer active.
884.	UDC (82) Eliminated linear search of COMMAND.PUB.SYS. for locating UDC files. The system directory now contains the location of the UDC files. Improved error detection and recovery in the :SETCATALOG command.(SMR # 7249,7597,7851)

FIX NUMBER	DESCRIPTION
885.	PROGEN (09) System UDC's will no longer be lost after system goes down and is brought back up.
886.	CI (51) The NOBUF FOPTION of FOPEN can now be overridden with a FILE equation.(SMR # 6121)
887.	SYSDUMP (01) SYSDUMP was corrected to prevent incorrect LDEV's from being printed in the error message "UNDEFINED classname USED AS OUTPUT DEVICE BY FOLLOWING DEVICES devicename". The problem occurred when adding a disc with an undefined output device.
890.	INITIAL (00) INITIAL will no longer get caught in a loop when the error "UNDEFINED CLASS (xxx) USED AS OUTPUT DEVICE BY FOLLOWING DEVCES" occurs. (SMR # 7448, 7531)
891.	INITIAL (00) INITIAL was fixed to allow the user to change the "MAX NUMBER OF SPOOLFILES KILOSECTORS" if the DISABLED LOGGING question is answered negatively. (SMR # 2319)
892.	INITIAL (00) This fixed the problem that occurred when the error message "DISC DRIVER DOES NOT EXIST" appeared even though the disc had been properly configured.
899.	FILESYS (50) FGETINFO will now return the name "$NULL" when the file name is requested for a file FOPENed as $NULL. Prior to this change, the file name was uninitialized (results were random).(SMR # 6676)
900.	FILESYS (50) FCONTROL to write EOF will now return FS ERROR 40 (access violation) if an attempt is made to write EOF to a mag tape, and if the file was opened read only. This applies to serial disc and SSLC devices also. (SMR # 7568)
901.	FILESYS (50) Corrected Fspace labeled tapes. (SMR # 7078,7642)
902.	OPCOMMAND (85) This fixes a JOBFENCE problem introduced when the new OUTFENCE features were implemented.

FIX NUMBER	DESCRIPTION
903.	ALLOCATE (54) Added logical procedures SFINDACTIVE (LDEV, DFIO) to ALLOCATUTIL.
903.	OPCOMMAND (85) ALTSPOOLFILE will now work on LDEV or active SPOOLFILES.
905.	LABSEG (86) Fixed the problem of writing over a labeled tape volume ID if no write-ring is on the reelswitch.
906.	OPCOMMAND (85) Corrected numerous :SHOWALLOW messages.
907.	IOCDRD0 (20) The card reader driver KATAKANA translation table wase modified to accomodate KATAKANA lower case characters.
908.	UTILITY (70) The intrinsic CTRANSLATE was modified to use the correct translation characters for KATAKANA lower case letters.
909.	SPOOLING (79) This fixes a bug introduced by the NOTIFY'OPERATOR procedure installed previously.
910.	OPCOMMND (85) This fix deleted ";DEFAULT" as a valid parameter in the :DOWNLOAD command.
911.	CATALOG Added messages for notifying operators in spooling and for CXDOWNLOAD in OPCOMMAND.
915.	INITIAL (00) Fixed the serial disc interface so that it won't detect EOF/EOT marks when attempting the forward space command. This caused the double EOF error message.
923.	SOFTWARE.CREATOR SOFTWARE.CREATOR will now correctly install or purge any communications products (MRJE & MTS) under any system configuration.
926.	HARDRES (55) This change will make consecutive calls to the timer consistant with each other. Time will not appear to move forward or backward in relation to successive timer calls.

FIX NUMBER	DESCRIPTION
929.	INITIAL (00) Configuration parameters will no longer be altered when changing memory size. (CONFDATA PROBLEM)
930.	SYSDUMP (01) Configuration parameters will no longer be altered when changing memory size. (CONFDATA PROBLEM)
933.	MESSAGE (59) Corrected hang of session after a =REPLY during a user break while waiting for a tape mount.
934.	SYSDUMP (01) Added IOCDPN0 entry back to SYSDUMP so it will reappear on the Series 33 as a system file.
935.	INITIAL (00) INITIAL will now print out memory resident in bank 0 upon launching the system to an operational state.
1000.	PVSYS (81) 1) VSUSER now can handle more than 10 users. 2) VSUSER now shows suspended jobs.
1001.	PCREATE (63) 1) Duplicate blocks of records no longer appear in the log files. 2) SF 30 no longer occurrs if the log process is awakened with an empty buffer to be written.
1007.	DEBUG (75) This fix has no effect on the DEBUG code. It is for documentation only. It outlines the breakpoint table and several of the subroutines of DEBUG.
1014.	CI (51) LISTF,2 now displays the correct number of sectors allocated to a file, instead of printing a negative value when this number is larger than 32767.
1018.	UDC (82) UDC's will now accept exclamation points within the body of the UDC for reasons other than replacing formal parameters. For example, to have COMMENT HI! printed with OPTION LIST, the body of the UDC should contain COMMENT HI!!. NOTE: If more than one exclamation mark is used, then a parameter will not be expected.

FIX NUMBER	DESCRIPTION
1019.	SOFTRES (56) Using MFDS and MTDS instructions on a data segment that is a stack could cause the real memory allocated to the segment to be lost if the process owning the stack was marked "OUT" and not changed to "ABSENT" when the segment was brought into memory. This has been fixed.
1020.	LABSEG (86) Corrected procedure CREATELTENT to handle VOLID of less than six characters on 2nd open of labeled tape.
1022.	STORE/RESTORE (52) SYSDUMP and STORE will now lock the file integrity SIR and re-read the file label of a file stored to tape before re-writing the file label in order to turn off the store bit. Previously, the file label was not re-read; store just wrote out its copy of the file label. There was no SIR protection used. This meant that, given the proper time sequence, the file label could be changed (say, by FCLOSE) but the change was not picked up by STORE (SYSDUMP). In some cases, an SF 102 resulted when an attempt was made to subsequently open that file (old copy of file label contained non-zero PCB VECTOR,i.e. invalid DST index).
1023.	UDC (82) A UDC command will not be allowed to exceed 16 characters in length. (If this does happen, then a SETCATALOG (CIERR 1951) will be returned.)
1024	INITIAL (00) Whenever INITIAL detects an error in classname, it now will return to step 3.2 if-- a "NO" response is given in step 3.83 or it-- any response is given in step 3.92.
1027.	FILESYS (50) FS ERR 70 will be returned if header/trailer spoolfile labels fail.
1027.	ALLOCATE (54) Changed PRIMEDEVICE to return false if header/trailer fails.
1027.	SPOOLING (79) The header/trailer procedure will return false if an I/O error occurs or if VFC is reset.

FIX NUMBER	DESCRIPTION
1027.	OPCOMMAND (85) The OPCOMMAND has been changed so that: 1. The operator cannot :DOWN the console LDEV. 2. The operator cannot switch the console to a :DOWNed or a pending :DOWNed LDEV.
1027.	CATALOG Messages were added for :DOWN :CONSOLE and header/trailer error #70
1028.	INITIAL (00) Now permits changing the input device during a reload of user files if not all the files were found on the first reel.
1029.	UCOP (07) UCOP now does no-wait I/O when writing out the message "CAN'T INITIATE NEW SESSIONS NOW". The system will no longer "hang" if I/O fails to complete.
1032.	CI (51) Fixed so that exceeding 28 continuation lines will not result in an error. The error message routine has been enhanced to give some definite information when a command extends over several lines.
1033.	CI (51) Break and timed reads are cleaned up after a RUN command just in case programs were to run with BREAK disabled and timed reads enabled; the exception is NOBREAK UDC's.
1034.	CI (51) A PURGEGROUP command with parameters will now remain intact for REDO.
1035.	INITIAL (00) INITIAL will now create one entry in the volume table for each configured disc.
1037.	CATALOG Messy message 2,3142.
1039.	SYSDUMP (01) Preped SYSDUMP with MAXDATA=32000 to provide sufficient space for expandable "DUMP FILE SETS" array.

FIX NUMBER	DESCRIPTION

1040. ININ (10)
DEBUG now handles COBOL II, two word instructions.

1047. OPCOMMAND (85)
A new procedure was inserted to find the absolute number of sessions & limits from CONFDATA.PV.SYS, The parms of the LIMIT command are then checked against those absolute limits.

1049. UDC (82)
Check for mixing of positional & keyworded parmmeters in the UDC command image.

1052. FILESYS (50)
1. Checks for overflow in the calculation of variable length records. If the record length is larger than allowed, an error will be returned.
2. Checks for overflow in the calculation of the total # of sectors. If the # sectors in the file overflow the field, an error will be returned.
3. Checks access type before allowing the return of unused unused space in FCLOSE. If the access is read-only and return of unused space is requested in FCLOSE, then an error will be returned.

1053. NURSERY (76)
During logon checking of IA/BA capabilities for session/jobs, the capability of the account is checked before that of the user. The CIERR message has been changed to indicate whether the account/ user was missing the required capability.

1055. DIRC (53)
Corrected a problem in which buffering of directory blocks was being mismanaged during a certain directory switching operation between non-system and system directories. Specifically, performing a PURGEACCT while a group was bound to its home volume set could result in posible corruption of either directory (subsystem/system).

1061. IOLPRT0 (19)
"0" and "-" CCTL specifications now observe the auto page eject condition.

1062. INITIAL (00)
"NOT ALL FILES FOUND- ANOTHER TAPE SET AVAILABLE?" has been changed to "nnn FILES NOT FOUND- ANOTHER TAPE SET AVAILABLE?"

FIX NUMBER	DESCRIPTION
1063.	INITIAL (00) This will prevent a halt without a message caused by a code segment being loaded over INITIAL's I/O program area and CST table; an "OUT OF MEMORY" message will now be printed.
1065.	LOADER1 (72) LOADPROC will now accept input names in lower case.
1066.	CI (51) REDO will no longer work after executing a nonredoable command.
1067.	CI (51) The Command Interpreter has been changed to: A. Dissallow :FILE equations on REPORT for list file. B. If an output file is new, it is closed as temporary.
1068.	CI (51) Allows for nested REMOTES in :FILE equations.
1073.	SYSDUMP (01) Resegmented and modularized the SYSDUMP code.
1074.	SPOOLCOMS (80) The jobs in INIT state will show as introduced by a :SHOWJOB command.
1075.	UDC (82) Allows for UDC's with a blank line as the first line of the body.
1083.	FILESYS (50) FCONTROL (2) and (6) have been corrected to work properly for variable-length record disc-files.
1084.	FILESYS (50) FILE SYSTEM has been corrected to properly handle an initial allocation of a negative number of extents. The initial allocation will be set to 1. Also, all FCB's will now go into the system file segment. Finally, FOPEN will now get the file system (multi access) SIR before searching the directory for an existing file label's address.
1085.	FILESYS (50) Corrected FILE SYSTEM procedure (FCONVBLK) to initialize a local variable which may be used (in a call to EXCHANGEDB) if an error occurs in reading or writing the file label.

FIX NUMBER	DESCRIPTION
1086.	FILESYS (50) FCLOSE will automatically write EOF when writing a MULTI-VOLUME unlabelled tape.
1090.	INITIAL (00) ULOGCOUNT for MANAGER.SYS is now set to one in order to inhibit PURGEUSER MANAGER.SYS.
1092.	INITIAL (00) Now provides recovery in the event of an error during a RESTORE of user files.
1093.	MMDISKR (57) Allows VDS to expand past 32k sectors to a max of 64k sectors.
1094.	DATASEG (67) Modifed SEGSTATE to make absent segments which may be concurrently accessed, so that processes do not get launched thinking that they are INCORE when they are not.
1095.	SOFTRES (56) Modifed MAKE ABSENT to search all of the READY and DISCARD lists for DST's to make absent.
1096.	CI (51) If unsuccessful, PURGEACCT on a PRIVATE VOLUME will not remove the account entry in COMMAND.PUB.SYS.
1097.	FILESYS (50) Bounds checking is now done on all parameters passed to FERRMSG.
1099.	CI (51) The Command Interpreter has been changed so that: (1) CXFILE will now take into account the NOLABEL parameters and turn a bit on in the PMASK. (2) LISTEQ2 will now correctly interpret the JDT information.
1101.	INITIAL (00) INITIAL will now do automatic speed sensing on the 26xx terminal on Series II/III. A carriage return is required on other types of terminals.
1102.	INITIAL (00) INITIAL will now reset the LOGOFF bit in the JMAT.

FIX NUMBER	DESCRIPTION
1105.	PVSYS (81) Corrected errors in setting private volume bits in the JIT entry which is used by the file system after a group is bound.
1109.	SYSDUMP (01) SYSDUMP will use FERRMSG to print FILE SYSTEM errors.
1110.	DEVREC (08) A :"JOB" command issued from within a JOB on an interactive device will now be recognized whether typed in lowercase or uppercase. (Before, lowercase would fail).
1116.	LABSEG (86) Changed the procedure CREATETLTENT to ensure that duplicate tapes are not assigned to the same PIN #.
1118.	INITIAL (00) Corrected the deletion of virtual disc space occupied by defective tracks.
1119.	INITIAL (00) Initialized the variables SYSTAPELDEV, SYSTAPETYPE, SYSTAPESTYPE if loaded from tape.
1120.	INITIAL (00) Increments HVOL when adding a new disc to the system domain.
1121.	INITIAL (00) Fixed WARMSTART to release the same amount of space as it allocates for JMAT,IDD,ODD.
1122.	INITIAL (00) Removed unnecessary code from the cold load device drivers.
1123.	INITIAL (00) Permit tracks to be deleted from an uninitialized pack being added to the System Domain.
1125.	CI (92) When the number of required UDC parameters was exceeded, the error message used to point to its last required parameter instead of pointing to the erroneous one. This has been fixed.

FIX NUMBER	DESCRIPTION

1126. UDC (82)
Previously, when a blank parameter resulted in a blank line, an unnecessary error message was generated. This has been corrected.

1127. UDC (82)
Corrections have been made to the way empty UDC's are handled (referring to UDC's which contain only the line defining the source and parameters of that UDC.)

1128. USER (83)
1). Previously, GENMESSAGE returned CCL for all errors. With this fix, the operation of GENMESSAGE will now conform to the documentation.
2). A new error number has been added to GENMESSAGE. If errnum=7, then the write to the destination file failed.

1129. HELPUSER (84)
HELP commands are now deblanked properly when the command is issued with two parameters.

1133. LABSEG (86)
If odd byte record size is specified for a labeled tape, then the record size in the label will be 1 byte larger because the 7970 does not write odd bytes.

1135. MORGUE (64)
The command sequence BREAK, ABORT on a process which has son(s) terminating at the time of the break will now abort properly instead of locking up the process structure (and hanging the terminal).

1136. LABSEG (86)
Will now correctly send AVR messages to the associated users.

1137. CATALOG
Will now correctly send AVR messages to the associated users.

1139. FILESYS (50)
Fixed for system failure number 353.

1140. PCREATE (63)
Previously, calling GETPRIORITY and specifiying "BM" caused a process to be incorrectly scheduled into the ES queue. This fix will cause the process to be scheduled correctly (linearly at pri=101).

FIX NUMBER	DESCRIPTION
1142.	CATALOG Messages were added for mounting and dismounting the disk (previously, such messages were program generated). Also, the message "PV RECOGNITION" was changed to be more informative.
1146.	ININ (10) After a power fail, the system clock interrupt rate will not be a fixed 100 ms. It will be the rate it was running if the clock interface was used.
1152.	SPOOLCOMS (80) Error recovery for SHOWIN/SHOWOUT now 1) treats the master console the same as any other terminal and 2) returns the correct error number in case of a non-syntax related error.
1155.	FILESYS (50) Changed FGETINFO to return the correct FOPTIONS for a spooled input device from a stream job.
1156.	FILESYS (50) Changed FCONTROL rewind so that the block transfer count is not zeroed out in EOF1 label for labeled tape.
1159.	OPCOMMAND (85) Fixed :STOPSPOOL command to take immediate effect by resetting active spoolfile and stopping.
1160.	STORE/RESTORE (52) Corrected RESTORE return value to the directory system while storing files and the private volume user is bound.
1161.	DEBUG (75) DEBUG will now display the full, printable ASCII character set.
1162.	SPOOLCOMS (80) Cleaned up "SHOWDEV" to prevent printing of superflous garbage for labeled tape.
1163.	PROCSEG (60) Allows BREAK and ABORT of a process structure when the USER-SON OF MAIN is in a mail wait.
1165.	PROGEN (09) Adds new datacomm console command, SHOWCOM.

FIX NUMBER	DESCRIPTION
1165.	CI (51) 1. Added command interpreter routines for IML. 2. Fixed CLINE misc array bugs.
1174.	CATALOG Load ERR 25 was changed from "DATA SEGMENT TOO LARGE" to "STACK SPACE REQUIRED EXCEEDS SPECIFIED MAXDATA".
1179.	FILESYS (50) Corrected fix to defaults of record size, record format, and blocksize in the label of labeled tapes.
1180.	LABSEG (86) Same as FILESYS fix #1179.
1187.	CI (51) Fixed a potential system deadlock (when the directory runs out of disc space and NEWACCT command is executed. (SMR # 10031)
1188.	STORE/RESTORE (52) Modified group name test to group and account name test to check for new logical device when restoring from tape containing intermixed PV and system domain files. (SMR #11076)
1189.	FILESYS (50) If the free disc space table becomes full in the middle of a PURGEGROUP or PURGEACCT command, the filename would still be left in the directory but only part of the file would remain allocated. Now if this happens, the file remains intact and the console operator is notified that the free space table is full. (SMR # 6032)
1189.	PVINIT (32) See description for FILESYS #1189.
1189.	ALLOCATE (54) See description for FILESYS #1189.
1189.	CATALOG See description for FILESYS #1189.
1190.	STORE/RESTORE (52) System failure #311 will no longer occur when restoring files with lockwords. (SMR #10225,10227,10275)

FIX NUMBER	DESCRIPTION
1191.	LOADER1 (72) Running a program file will no longer alter its last change date in the file label in B.01.00. (SMR # 10316)
1192.	PVSYS (81) Fixed a special case of foreign disc recognition.
1193.	PVPROC (31) Fixed a special case of foreign disc recognition.
1194.	SYSDUMP (01) SL changes - The default for installing patches into the system SL will now be "0". This means that the patch area maintenance word will be appended to the code segment.
1195.	SYSDUMP (01) Fixed message "IS classname A FOREIGN DISC?".
1197.	PVSYS (81) Corrected an error produced when the operator entered two logical mounts making the private volume inacessible to further mounts or dismounts until Coolstart was performed. (SMR # 10089)
1209.	SYSDUMP (01) Prints descriptive error messages when encountering an ATTATCHIO error.
1211.	SYSDUMP (01) Prevented SYSDUMP from not dumping the directory when it is completely full.
1212.	FILESYS (50) Fixed problems with FGETINFO that would occasionally cause system failure 9 when called on a $NULL file.
1213.	USER (83) Fixed GENMESSAGE intrinsic. When the first line of a message in the message catalog is all blanks or when the first line contains a "%" as the only non-blank character, the first line printed on the output device will be a blank line. Previously, the message printed on the output device would start with the second line.

FIX NUMBER	DESCRIPTION
1223.	DISPATCH (04) ADDTOWS call now performed for processes quiesced while using an extra data segment.
1236.	CATALOG Added error messages for SHOWCOM. Changed CS I/O error so that it picks up the error number.
1244.	CI (51) Changed call to the message system for non-existent program files (a missing message of sorts).
1247.	PVPROC (31) Fixed bug in Foreign Volume recognition involving remounts of the in-use disc.
1248.	NRIO (62) Reset SYSDB for invalid devices in device status, else system fail.
1250.	PCREATE (63) Fixes an ES subqueue problem introduced by over-writing the variable declaration in the original change file.
1253.	DATASEG (67) Assured that the ADDS 0 instruction will not stack overflow with negative value in relation to maxdata.
1255.	COMM'INT (51) Set up CKBUILD entry point so it doesn't use parse only flag.
1257.	SPOOLCOMS (80) Fixed INITIATESPOOLER so that it doesn't zero-out of a spooler process in which ADOPT has built stack markers.
1258.	OPCOMMAND (85) Fixed ADOPT so that it can handle a process with open standard files.
1259.	ALLOCATE (54) When magnetic tape devices are closed, the session will no longer hang until rewind occurs.
1261.	FILESYS (50) Now checks for out of range addresses for nobuf reads/writes of a non variable file where the requested starting record number is past the EOF for a read or past the file limit for write.

II. SUPPORTED UTILITIES

A. UTILITIES MODIFIED

UTILITY	LEVEL
*ASOCTABL	-- --
*DISKED2	01.01
*DPAN2	01.01
*FREE2	01.01
*IOCDPN0	-- --
*LISTDIR2	01.01
*LISTEQ2	01.01
*LISTLOG2	01.01
PATCH	01.00
*MEMLOGAN	01.01
*MEMTIMER	01.01
*SADUTIL	01.01
*SLPATCH	01.00
SPOOK	01.01
RECOVER2	01.00

* INDICATES UTILITY UPDATED/CHANGED BY THIS M.I.T.

B. ENHANCEMENTS

FIX NUMBER — DESCRIPTION

897. SPOOK
   A. Changed commands ALTER, COPY, APPEND for:
      1. (DFID,DFID,...
      2. USER.ACCT
      3. @.@
   B. Spook can now be run in batch mode.

1041. ASOCTABL
   Added the ability to list the current ASSOCIATE file via :RUN ASOCTBL,LIST.

1081. DPAN2
   Eliminated the printing of the octal portion of the PCB table prior to the formated portion of that table.

1082. DPAN2
   Reformatted the terminal buffer area in DPAN so that the entire ASCII representation of the buffer is displayed on a single line.

FIX NUMBER	DESCRIPTION

1114. DPAN2
Eliminated all entry points except for EIGHTLPI. (Default is SAVEPAPER.)

1182. LISTLOG2
The output listing file (formal designator LOGLIST) is now opened NEW instead of in the temporary domain.

C. CORRECTIVE SOFTWARE CHANGES

809. LISTEQ2
Changed LISTEQ2 so that file equations with the parameter NOLABEL are printed. (SMR # 5628)

811. LISTDIR2
Modified the REC SIZE and BLK SIZE values for files with variable length records to correspond with the values printed by LISTF. (SMR # 6731)

817. DISKED2
Invalid parameters no longer cause "BOUND VIOLATION" aborts on the Series II/III and "STACK UNDERFLOW" on the Series 33.

836. LISTLOG2
LISTLOG2 no longer displays the field "LINE LIM" of log record type 2 (job initiation). The value was meaningless.(SMR # 6934)

864. LISTDIR2
Changed the LISTDIR2, LISTF command so that FOPTIONS now include "KSAM". LISTF with the MAP option now correctly handles files with 32 extents. (SMR # 7082,7082)

898. SADUTIL
SADUTIL will now work with the HP2888 disc. The magtape will now skip over bad spots on the tape. (SMR # 7879,2654,2840)

924. IOCDPN0
Corrected bugs on the Series 33. Miner bug fixes also on Series II/III. Restored DR&# command. Floppy verify was fixed on the Series 33.

1013. DISKED2
Disked2 will no longer enter into an endless loop printing "ERROR" when a colon is typed in as the first character in the command line. Instead, it will now exit to the CI.

FIX NUMBER	DESCRIPTION
1057.	DPAN2 Eliminated bounds violations caused by invalid dump tapes. Such tapes can no longer cause integer overflows.
1077.	DPAN2 This correction allows for the proper formatting of the Series 33 registers when other than the normal entry point is used in DPAN2.
1078.	DPAN2 This correction eliminates the possibility of an infinite loop while printing pins that are impeded by a SIR. It also insures the printing of all impeded pins.
1080.	DPAN2 Disallowed the printing of the formatted tables when the DST,CST and PCB pointers are all invalid.
1098.	LISTEQ2 Modified to correctly interpet JDT information regarding the LABEL and NOLABEL parameters.
1138.	DISKED2 Modified to dump both upper and lower case ASCII.
1139.	DISKED2 Modified to dump both upper and lower case ASCII.
1151.	LISTDIR2 For variable length files, there are now two extra words/block. The LISTF command will now list those words when reporting the recordsize.
1153.	DPAN2 DPAN2 now decodes the status register correctly on the formatted registers page.
1154.	DISKED2 Internal documentation modified to reflect code function. Parameter checking in DUMP extended. Second DUMP parameter (number of sectors dumped) defaults to 1.
1158.	LISTEQ2 The size of the JDT had been changed for the 1918 IT. Flowover LISTEQ2 had not been changed to accommodate the new larger size of this table. Consequently, the messsage xxfailure to get jdtxx was printed by LISTEQ2. This has been fixed.

FIX NUMBER	DESCRIPTION

1164. FREE2
FREE2 will now give accurate reports when the output is redirected via a :FILE equation (FREE2 will not report its own output spoolfile). Note that FREE2 when submitted as a job, will report the spool files created by the job.

1166. LISTLOG2
Changed so that:
1. All error messages which are the result of File System errors will now include the file system error number message, "ERROR NUMBER yy".
2. A new message has been added to tell the user when a logfile is an empty file.

1167. LISTLOG2
LISTLOG2 will now abort without excuting if the user does not have SM capability.

1168. MEMTIMER
If the user does not have system manager capability, MEMTIMER will now terminate without executing.

1169. MEMLOGAN
If the user does not system manager capability, MEMLOGAN will now terminate without executing.

1170. DPAN2
Changed DPAN2 so that the current PCB stack markers are traced when the system halt occurs while on the ICS.

1181. LISTLOG2
Whenever an FWRITE error occurs to the output file LOGLIST, the program will terminate with QUIT parameter 99 after printing an error message on the standard listing device.

1189. SADUTIL
See description for FILESYS #1189.

1199. DPAN2
Changed so that:
A. Prints correct controller status and controller error status when Unit 0 is not configured. (SMR # 5773)
B. Prints all DS DIT's. (SMR # 10381)
C. Supports the full, printable ASCII character set when printing the terminal buffers. (SMR # 10617)

FIX NUMBER	DESCRIPTION
1205.	LISTDIR2 "LISTF" of "LISTDIR2" was fixed so that it correctly reflects the access flags of a file currently being "RESTOREd".

SEGMENTER HP32050A.01.02

N00N050A.HP32050.SUPPORT

CORRECTIVE SOFTWARE CHANGES

807. SEGPROC (02)
The loader system SL directory cache is cleared when SL.PUB.SYS is specified in the SEGMENTER SL COMMAND.

1015. SEGUTIL (71)
SEGMENTER command 'CLEANUSL' will no longer get an error #94 (unexpected end of file) on USL's generated by the "FORTRAN" compiler.

1017. SEGDVR (03)
Corrected spelling error in heading.

1026. SEGUTIL (71)
CLEANUSL command has been fixed so that an invalid son pointer no longer results when:
(1) All RBM's are inactive in a segment.
(2) All secondary entry points of an RBM are inactive. An invalid segment list pointer results when all segments of a USL are inactive.

1031. SEGPROC (02)
Two COPYSL's or CLEANSL's in a row resulted in the first SL not being saved. This has been fixed.

1124. SEGPROC (02)
The problem where SEGMENTER gets caught in a loop while processing error #67 "Common declared with different size" has been fixed.

1141. SEGPROC (02)
SEGMENTER's command "PURGERBM" did not unlink the sons of a purged entry in the directory example. It is sitll possible to find procedures of a purged segment.

1145. SEGPROC (02)
Segmenter's "COPY SEGMENT" command will no longer cause a CPU loop if an empty segment is copied into a USL where a segment by the same name already exists.

A. ENHANCEMENTS

1. A new $CONTROL option, DEFINE, has been added. When invoked, the DEFINE option causes the body of defines to be written out to disc rather than kept in the memory-resident symbol table. This is useful in large programs where a symbol table overflow might otherwise occur.

   The option must be invoked before any defines are declared, and once invoked can not be revoked. The file containing the define bodies is an SPL scratch file and is deleted at the end of the compilation.

2. A $INCLUDE option has also been added, to permit the inclusion of text files into the SPL source. The format of the statement is:

   $INCLUDE <filename>;

   where <filename> is the fully qualified name of the file to be included. The included file may contain other $INCLUDE statements, to a maximum level of 10. The file is included immediately following the statement. If the included file is numbered, its line numbers are used; if unnumbered, blanks are inserted. No source merging is done on the included text.

3. When the MAP $CONTROL option is used, reference PARAMETERS to procedures will now be flagged with an "(R)".

B. CORRECTIVE SOFTWARE CHANGES

1. SR # 2564 - The initialization of a pointer to an external address was not being done correctly. This has been fixed.

2. SR # 2615 - The amount of Primary DB space available was incorrectly assumed to be 801 words. The correct figure of 908 words is now used.

3. SR # 3465 - When a PB-relative array is not fully initialized, only the part initialized is allocated and no error message was output. A warning is now output.

4. SR # 3545 - USL files created by the SEGMENTER BUILDUSL command produced an end-of-file I/O error when used in a compile. This has been remedied.

5. SR # 3923 - Incorrect parameter stacking occurred when one of the parameters in an OPTION VARIABLE procedure was a label. The parameters are now stacked correctly.

6. SR # 4037 - Long variables used as parameters are handled incorrectly when computation is done in the procedure call. The correct code is now generated.

7. SR # 4104 - INNERLIST listed extraneous instructions when byte pointers or arrays were compared. The extraneous instructions are no longer listed.

8. SR # 4178 - The absolute value of long array elements was improperly calculated. The correct value is now returned.

9. SR # 4214 - Any misspelling of a reserved word in a subprogram compile terminated the compilation with no error message. An error message is now generated.

10. SR # 4356 - A define used as an entry point was improperly handled. This now works correctly.

11. SR # 4436 - Procedure calls as parameters malfunctioned. They now work properly.

12. SR # 4608 - No warning message was generated when a constant was passed as a reference parameter. A warning message is now generated.

13. SR # 5564 - An intrinsic with a one-character name in a user-defined intrinsic file was not being referenced correctly. The intrinsic is now accessed properly.

14. SR # 5789 - ENTRY points used inside the procedure in which they are declared were not referenced properly. This now works correctly.

15. SR # 5801 - An array could be equivalenced to itself with no error message. This is no longer permitted.

16. SR # 5947 - This SR was only partially corrected in Version 07.02. $TWENTY & $THIRTY now work correctly in all cases.

17. SR # 6182 - When a long pointer passed by reference was used on the left-hand side of an assignment statement, incorrect code was generated. The correct code is now generated.

18. SR # 7086 - LOGICAL variables with values greater than 32767 were incorrectly converted by the REAL type transfer function. The conversion is now done properly.

19. SR # 7174 - If an initialization list for a direct byte array contained a repetition factor with multiple elements, an error message was generated. The list is now handled properly.

20. SR # 7561 - Incorrect code was generated for a byte index into an array. The proper code is now generated.

21. SR # 7698 - Multiple assignments of long array elements with a long constant resulted in incorrect code. The correct code is now generated.

22. SR # 7769 - Incorrect code was generated when the parameter to the LONG type transfer function was a call to a double subroutine. This has been rectified.

23. SR # 7804 - An incorrect SDEC was generated for the ASSEMBLE (MVBW AS) statement. The correct SDEC is now generated.

24. SR # 7825 - Paging was incorrect when a listing was placed on disc and then FCOPYed to the printer. The paging is now correct.

25. SR # 8031 - If more than 1003 USL type 1 headers were generated the compilation aborted without notice. An error message is now produced and the compilation terminated.

26. SR # 8184 - A bounds violation occurs under certain conditions when a label is declared, but not used. This has been fixed.

27. SR # 8348 - The use of a double integer as a shift-count wasn't flagged as an error. This is no longer permitted.

28. SR # 9341 - The $TWENTY option was ignored when code was generated for long parameters passed by value, resulting in erroneous loads and stores. The correct code is now generated.

29. SR # 9546 - When SPL output the "string too large" error message, the compilation was aborted instead of recovery being attempted. Recovery is now attempted.

30. SR # 9550 - If an element of a virtual array of type real or double was assigned a value, the X register, and in some cases the stack, may have been corrupted. This no longer happens.

31. SR # 9567 - If an attempt to expand the USL directory failed, the hash links may have been corrupted and the USL file rendered unusable. This has been fixed.

32. SR # 9625 - If the character following the final END in a program is not a period, no error message is generated. An error message is now output.

33. SR # 9857 - When an error occured in a subprogram compilation, the compiler would, under certain conditions, corrupt the hash links in the USL file, rendering it unusable. This has been corrected.

    NOTE: This SR is similar to, but not the same as, SR# 9567.

34. SR # 9882 - Under certain very specific conditions, SPL would overlay code with data. This no longer occurs.

35. SR # 10105 - If an S - relative variable was used as an index to a direct array which is a parameter in a procedure call contained in a subroutine, erroneous code was generated. The correct code is now generated.

36. SR # 10161 - When an integer procedure with a parameter of type double, real, or long was used to index a pointer or array which is passed as a parameter, erroneous code was generated. This no longer happens.

37. SR # 10182 - Procedures were allowed as value parameters. An error message is now generated.

38. SR # 10231 - When the $CONTROL ADR listing option was invoked, the occurrence of an OWN long varaible declaration with a non-zero fractional part would result in a bounds violation. This has been remedied.

39. SR # 10303 - A code sequence of the form:

    ```
    1) <logical var.> := A<=B<=C;
    2) IF <logical var.> = TRUE THEN . . . ;
    ```

    didn't work. The sequence now works properly.

40. SR # 10862 - The documentation does not make clear that for a statement of the form:

    ```
    MOVE * := "<string constant>";
    ```

    the stacked address is assumed to be a byte address. The documentation will be changed to reflect this.

41. SR # 11178 - If a conditional logical equivalence required more than 10 code inserts, a bounds violation occurred. This no longer happens.

42. SR # 11277 - Double to long conversion in a subroutine produced incorrect S-relative offsets. The correct offsets are now produced.

43. SR # 11279 - Incorrect code was generated for the statement:

    ABSOLUTE(<constant>).(<partial field>) := TOS;

    Correct code is now produced.

BASIC/3000 HP32101B.00.12

NOON101B.HP32101.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

1. SR #10368.
   The error message "LOADER ERROR #40", "Too many procedures loaded", while running a program in the BASIC Interpreter, may be the result of a problem in MPE. The SR has been referred to MPE for resolution.

2. SR #10397.
   When trying to extract the decimal value from a real number, the Interpreter sometimes incorrectly integerizes this number. The same results occur in the compiled version of the program. This situation may be the result of floating-point numeric representation in the 3000, and the SR has been referred to Hardware Engineering.

3. SR #11871.
   When an apostrophe (') appeared within a quoted string within an image for PRINT USING, the apostrophe was printed out as well as a zero (0). For example, the following program produced erroneous results:

   ```
   >10 IMAGE "'S"
   >20 PRINT USING 10
   ```

   The result printed was 'OS.

   This has been corrected.

CORRECTIVE SOFTWARE CHANGES

1. SR# 3156 - The use of the CROSSREF ALL option caused the compiler to abort in some very rare conditions when it was printing out the label cross reference. This problem has been corrected.

2. SR #7114 - Under rare conditions, the compiler would generate incorrect code for logical IF statements. The compiled program would then go into an infinite loop during execution. This problem has been fixed.

3. SR #7398 - On rare occasions, a constant pool was placed in code used for displaying a character constant. The resulting code would produce incorrect output. This has been fixed.

4. SR #7452 - If $CONTROL NOWARN was in effect, the summary of errors and warnings at the end of a program would state that no warnings had occurred. The compiler has been changed to emit the number of warnings that were suppressed instead of NO WARNINGS.

5. SR #7459 - In a subroutine, if a simple variable appeared with a subscript, the error message "SUBROUTINE USED AS PRIMARY" was given incorrectly. The compiler has been changed to give the error message "SIMPLE VARIABLE HAS SUBSCRIPT" in this case.

6. SR #7497 - In nested function calls, if one of the inner functions returned a character string longer than 126 characters, the compiler generated incorrect code. This would cause incorrect program results. Valid code is now generated.

7. SR# 7537 - Character string literal parameters totaling more than 126 characters caused other literal parameters in a subroutine call to be passed incorrectly. All literal parameters are now passed correctly.

8. SR #7668 - The use of an undefined variable as a subscript on the left side of an assignment statement was not properly flagged with a warning by the compiler. The message "REFERENCED VARIABLE NOT DEFINED" is now given in this situation.

9. SR #7732 - The FORTRAN compiler destroys some contents of a USL file if it is necessary to expand the directory. The problem appears when the USL is prepared, with "ERROR #1 ILLEGAL HEADER". The problem has been corrected.

10. SR# 8180 - Secondary Entry Points to function subprograms that were (recursively) referenced within that subprogram caused the compiler to abort or go into an infinite loop. This problem has been corrected.

11. SR# 8200 - A DOUBLE PRECISION composite number with a total number of bits greater than 48 (3 words) produced an error #75 "PACKED TYPE OVERFLOW". When the total number of bits was less than or equal to 48, the high order word contained unpredictable values. DOUBLE PRECISION composite numbers are now constructed correctly.

12. SR# 10612 - Program units that initialized large amounts of storage with DATA statements could cause the compiler to abort with an END OF FILE ON FTNUT2. The file is now large enough to correct this problem.

13. SR# 10640 - Alternate Returns from subprograms occasionally had incorrect code generated for branches to alternate return labels. This caused unpredictable results when the program was run. This problem has been corrected.

14. SR# 11118 - LOGICAL IF statements containing another LOGICAL IF statement were not flagged as errors. This caused code to be generated that produced unpredictable results when the program was run. An error message #199 IMPROPER DEPENDENT STATEMENT is now issued.

BASICOMP/3000 HP32103B.00.12

NOON103B.HP32103.SUPPORT

CORRECTIVE SOFTWARE CHANGES

SR #9587.
When the invoked program name was entered into the BASIC program as a lower-case name, the name was never upshifted. This resulted in an "unresolved program external" at prep time. This problem has been corrected.

A. ENHANCEMENTS

1. SR #5579 When User Indicators (U1 - U8) were used for file conditioning (columns 71 - 72 of the File Specifications), the compiler diagnostics listed them as unreferenced indicators. But actually, they are being referenced for file conditioning. This incorrect diagnostic is no longer reported. Instead, the Cross Reference option will report them as Defined and Referenced on that line.

2. SR #6491

```
*********************************************************
*                                                       *
*     A VARIETY OF IMAGE LOCKING MODES FROM RPG         *
*     INCLUDING ENTRY (RECORD) LEVEL LOCKING HAS        *
*                    BEEN IMPLEMENTED                   *
*                                                       *
*********************************************************
```

This enhancement provides you with 5 new Image unconditional locking modes from RPG to satisfy most of your interactive transaction processing needs by improving data processing throughput while maintaining data integrity. The five new locking modes are:

1) Lock the entire Image data base during the complete execution of the program [data base locking for duration].

2) Lock an Image data set during the complete execution of the program [data set locking for duration].

3) Lock a record whenever that record is accessed [record level locking].

4) Lock the Image data set whenever a record from that data set is accessed [data set locking per record].

5) Enable locking so that user supplied routines (probably SPL) may perform all locking and unlocking.

Prior to this enhancement, the only locking mode available was to lock the entire IMAGE data base whenever a record from that data base was accessed [data base locking per record].

Following are detailed descriptions of the new locking modes. Also, discussions on locking precedence, warnings, and programming cautions are presented.

a. Data Base Locking For Duration

Locking the data base for the duration of the executing program allows only you to access the data base. For example, if you are reading the data base to produce a report, all the current data values are frozen and cannot be unknowingly modified by another program. This ensures that your produced report is accurate with the data you expected. Also, suppose you wish to update the entire data base before other programs use it to produce reports. In this case, you may lock the entire data base to prevent any reports to be generated, then update it, and finally unlock it so that the other programs may now generate reports from the proper updated data values.

To request RPG to lock the entire data base for the duration of the executing program, you simply put a "B" in column 66 as the Image open mode on the KIMAGE specifications (see pages 4-34 to 4-35 of the RPG Reference Manual for an explaination of KIMAGE and open mode). The data base name that appears on the same KIMAGE record specifies the data base to be locked.

Specifically, locking and unlocking will be automatically performed as follows:

> Lock the entire data base at initialization time (before entering the RPG logic cycle) and unlock the data base when the program terminates.

EXAMPLE:

```
KIMAGE DBNAMEB
^
column 53 of the File Specifications
```

KIMAGE specifies "DBNAME" to be the Image data base name. The "B" following DBNAME requests RPG to unconditionally lock the data base for the duration of the executing program.

b. Data Set Locking For Duration

Data set locking for duration is equivalent to data base locking for duration as presented above, except that a data set is locked for the duration of the executing program.

To request RPG to lock at this mode, you put an "S" in column 66 as the Image open mode on the KIMAGE specifications. The corresponding file name in columns 7 - 14 of the File Description Specifications is the data set to be locked. If a DSNAME record is specified, then the name given on the DSNAME record is the data set to be locked, overriding the name given in columns 7 - 14 (see page 4-37 of the RPG Reference Manual for an explaination of DSNAME records).

Locking and unlocking will be performed as follows:

> Lock the entire data set at initialization time (before entering the RPG logic cycle) and unlock the data set when the program terminates.

EXAMPLE:

```
KIMAGE DBNAMES
KDSNAMEDSET
```

KIMAGE specifies "DBNAME" to be the name of the IMAGE data base. The "S" following DBNAME specifies data set locking for duration. KDSNAME specifies "DSET" to be the data set name that will be locked.

c. Record Level Locking

Record locking will significantly improve interactive transaction processing performance. When accessing an Image record entry by item value, you can request RPG to only lock that record which prevents further access to it until it is unlocked. The other records with different item values will still be accessible for processing.

To request RPG to lock at the record level, you simply put a "R" in column 66 as the Image open mode on the KIMAGE specifications (see pages 4-34 to 4-35 of the RPG Reference Manual for an explanation of KIMAGE and open mode).

Record level locking only may be requested when the Image Input/Output Mode Field is either 5, 6, C, or R (column 67 of the KIMAGE specifications). This is neccessary since RPG must be given an item value to determine which record to lock (RPG locks the record which has this item value). KITEM must also be specified when describing the Image data base in RPG.

If record level locking is requested but the Input Output Mode Field is not 5, 6, C, or R then the compiler will issue Warning #274:

```
"INPUT/OUTPUT MODE MUST BE 5,6,C, OR R FOR
 RECORD LEVEL LOCKING, ASSUME 1 IN COLUMN 66."
```

In this case, the compiler assumes data base locking per record (1 in column 66 of the KIMAGE specification) which is explained in section d.

Locking and unlocking will be performed as follows:

1. For an input file:
   Each record is locked before it is read, and unlocked after it is read.

2. For an output file:
   Each record is locked before it is written, and unlocked after it is written.

3. For an update file:
   Each record is locked before it is read and unlocked after it is updated. (If not updated, the record is unlocked before the next record is read.)

EXAMPLE:

```
KIMAGE DBNAMERC
KITEM  INAME
KLEVEL
KDSNAMEDSET
^
column 53 of the File Specification
```

KIMAGE specifies DBNAME to be the data base name. The "R" specifies record level locking. The "C" following the "R" specifies Chained Sequential Read. KITEM specifies INAME to be the item name used in referencing the data base. KLEVEL defines passwords. KDSNAME specifies DSET to be a data set name.

d. Data Base Locking Per Record

This is not a new enhancement and has been the only locking mode available until now. A brief description of this locking mode will be presented to clearly illustrate its effects and its differences with the other new locking modes.

Data base locking per record instructs RPG to lock the entire data base whenever a record from that data base is accessed. To request this mode of locking, you put a "1" in column 66 as the IMAGE open mode on the KIMAGE specifications.

Locking and unlocking will be performed as follows:

1. For an input file:
   The data base is locked before each record is read, and unlocked after it is read.

2. For an output file:
   The data base is locked before each record is written and unlocked after it is written.

3. For an update file:
   The data base is locked before each record is read, and unlocked after it is updated. (If not updated, the data base is unlocked before the next record is read.)

e. Data Set Locking Per Record

Data set locking per record is equivalent to data base locking per record, except that the data set is locked whenever a record from that data set is accessed.

You may request RPG to lock at the data set level per record by putting a "9" in column 66 as the IMAGE Open Mode on the KIMAGE specifications. Any Input Output Mode may be specified on column 67 of the KIMAGE specifications for this locking level.

The corresponding file name entered in columns 7-14 of the File Specifications is the name of the Data Set that will be locked. If a DSNAME record is used, then the name entered on the DSNAME record is the name of the Data Set to be locked, overriding the file name given in columns 7-14. (See page 4-37 of the RPG Reference Manual for an explanation of DSNAME records.)

Data Set level locking per record will be performed as described above for data base locking per record.

EXAMPLE:

```
KIMAGE DBNAME9
KDSNAMEDSET
```

KIMAGE specifies "DBNAME" to be the name of the Image Data Base. The "9" after DBNAME specifies Data Set level locking. The "blank" after the "9" specifies

that the Image I/O mode is write output. KDSNAME specifies "DSET" to be the name of the Image Data Set that is to be locked.

f. Enable Locking

This locking mode instructs RPG to open the Image data base with locking enabled. RPG will not automatically lock and unlock the data base at any level. Instead, you may supply your own routines (most easily written in SPL) to handle all locking and unlocking for you. This gives you complete control over Image locking facilities.

To request RPG to enable locking, you put an "L" in column 66 as the Image open mode of the KIMAGE specifications. The data base name specified on the same KIMAGE record is the IMAGE data base to be opened with locking enabled.

EXAMPLE:

```
     KIMAGE DBNAMEL
```

KIMAGE specifies "DBNAME" to be the IMAGE data base name. The "L" following DBNAME instructs RPG to open "DBNAME" with locking enabled.

g. Locking Precedence and Warnings

If your program defines the same data base and/or data set more than once, then all locking modes specified in the definitions for that data base and/or data set must be the same as the very first locking mode specified. For example, if a data base is first defined with data base locking for duration and later defined as record level locking, then this is a logical error because there is no reason to perform record level locking on the data base when the entire data base is already locked for the duration of the program. The compiler will issue a warning and default the record level locking to the first locking mode specified (data base locking for duration).

The locking modes have the following precedence among each other:

```
     Highest precedence --> B
                            S
     Lowest  precedence --> 1, 9, R, L
```

When an error occurs because all locking modes are not the same as the first locking mode specified for a data base, the compiler will perform either of the following:

1) If the first locking mode specified for a data base is of higher precedence than all of the following locking modes specified for the same data base, then the compiler will issue a warning and default all lower precedence locking modes to the first mode specified.

2) If the first locking mode specified for a data base is of lower precedence than any of the following locking modes specified for the same data base, then the compiler will issue a terminal error.

The compiler will not default a higher precedence locking mode to a lower one because this will not ensure data security. For these cases, the compiler will issue one of the following terminal error messages (278T, 279T):

"LOCKING MODE INCONSISTENT, FIRST MODE IS S, THE FOLLOWING MODE IS B."

"LOCKING MODE INCONSISTENT, MODES FOLLOWING FIRST LOCK MODE ARE OF HIGHER PRECEDENCE."

For example, if a data base is first defined with data set locking for duration and later defined with data base locking for duration, then this results in terminal error #278 and the second lock mode is not defaulted to the first.

In general, the compiler enforces the following algorithm on programs which define the same Image data base and/or data set more than once.

For the first occurrance of a unique named data base...

If locking mode B is specified, then ensure all other definitions of the same data base also specify locking mode B. If the lock modes are not all B then assume B and issue Warning #275:

"LOCKING MODE INCONSISTENT FOR DATA BASE, FIRST MODE IS B, MODES AFTER ARE DIFFERENT, ASSUME B."

Else if locking mode S is specified, then ensure all other definitions of the same data base and data set name specify locking mode S. If the lock modes are not all S then assume S and issue Warning #276:

"LOCKING MODE INCONSISTENT FOR DATA SET, FIRST MODE IS S, MODES AFTER ARE DIFFERENT, ASSUME S."

Else if locking mode 1, 9, R, or L is specified, then ensure all other definitions of the same data base and data set name specify 1, 9, R, or L respectively. If the lock modes specified are not all 1, 9, R, or L respectively the issue Warning #277:

"LOCKING MODE INCONSISTENT, MODES FOLLOWING FIRST LOCK MODE ARE DIFFERENT, ASSUME FIRST MODE."

h. Programming Cautions

1. If your program is to execute with more than one outstanding lock then you must have Multiple RIN Special Capability. Extreme caution should be taken to prevent deadlock when executing with Multiple RIN. (See page D-1 of the IMAGE Reference Manual for an explaination of MR capability.)

2. If your program defines 2 or more different IMAGE data bases, but you use the MPE FILE Equate Command to refer them all to the same actual data base, RPG will not know about this FILE Equate. RPG behaves like it is accessing many different data bases. Thus, if you wish to lock all the data bases specified in your program, you must have Multiple RIN Special Capability, even though all the data bases have been equated to just one actual data base.

3. See pages 4-13 to 4-13g of the IMAGE Reference Manual for information on locking levels, locking strategy, and choosing a locking level.

i. Example of Defining an IMAGE Data Base and Locking Mode in the File Description Specifications.

```
FISDATA1 IC  F      72R14AI    51 DISC
F                                        KIMAGE DBNAMER5
F                                        KITEM  DBITEM
F                                        KLEVEL
F                                        KDSNAMEDSET
FISDATA2 IC  F      72R14AI    51 DISC
F                                        KIMAGE DBNAMERC
F                                        KITEM  DBITEM
F                                        KLEVEL
F                                        KDSNAMEDSET
FISDATA3 IC  F      72R14AI    51 DISC
```

```
F                                                KIMAGE DBNAMERR
F                                                KITEM  DBITEM
F                                                KLEVEL
F                                                KDSNAMEDSET
```

j. Summary of Locking Modes

COLUMN 66 OF KIMAGE	MEANING
B	Lock data base for duration of program execution.
S	Lock data set for duration of program execution.
1	Lock data base whenever a record from that data base is accessed.
9	Lock data set whenever a record from that data set is accessed.
R	Lock record whenever that record is accessed.
L	Open data base with locking enabled so that user supplied routines may perform all locking and unlocking.

3. SR #6538 When running a job stream which uses User Indicators (U1 - U8), the User Indicator settings were not printed in the job stream listing. Now, the User Indicator record is echoed onto $STDLIST to inform you of each Indicator's value.

4. SR #6683 A device name specified on the File Card may now be preceeded with blanks. The compiler automatically strips off leading blanks, left justifies the name in the field, and issues Warning #272:

   "DEVICE NAME NOT LEFT JUSTIFIED IN FIELD, ASSUME JUSTIFIED."

5. SR #6717 RPG did not detect when a program attempted to output to a new file with a file name that already existed. Now, RPG will runtime abort with a Fatal File Error if this situation occurs. A File Information Display is also printed with ERROR NUMBER 100:Duplicate File Name.

6. SR #7231 RPG did not verify that a file specified to be a KSAM file was indeed a KSAM file. RPG now makes this verification and if it fails, will abort at runtime with a Fatal File Error, giving the added message:

   "FILE SPECIFIED AS A KSAM FILE ON FILE DESCRIPTION CARD IS NOT A KSAM FILE."

7. SR #7254 When specifying IMAGE characteristics on the File Description Card, no error message was given when the characteristics were incomplete or inconsistent. RPG now issues appropriate error messages.

8. SR #7862 For fixed length records, if the actual record length is less then the logical rcord length specified on the File Description Card, RPG will halt with the runtime error message:

   "UNIDENTIFIED RECORD...ACTUAL<LOGICAL RECORD LENGTH SPECIFIED ON F CARD."

   This is detected at the initialization phase of the RPG cycle and so it appears only once. You may respond or prerespond to this with 0 to 5. If you respond with a "0", this continues normal RPG processing, and the error message is essentially just a warning.

   To prerespond to this error, put a 0 to 5 in column 56 of the Header Specification Card.

   WARNING: If you respond with a "0" to continue processing, since the actual record length is less than the logical record length, RPG will pad the extra record length with zeroes (in numeric fields) and blanks (in alpha fields

   This recoverable error was implemented to ensure that you do not unknowingly input records whose actual length is shorter than the length specified in the program.

9. SR #9307 Error 6T (Header space overflow in USL file) was given incorrectly by RPG when PARM operations did not immediately follow an EXIT operation. Now, RPG issues Terminal Error #674:

   "PARM OPERATIONS MUST IMMEDIATELY FOLLOW AN EXIT OPERATION."

10. SR #9390 The compiler did not issue a warning if there was no actual Line Counter Specification Card even though it was specified on column 39 of the File Description Card. The compiler now issues Warning #413:

"NO LINE SPECIFICATIONS FOR FILE WITH L IN COLUMN 39 OF FILE SPECIFICATION."

11. SR #10471 The RPG compiler did not ensure that the File Description Specification column 28 contained an "R" (Random processing) if column 16 contained a "C" (Chained access). The compiler will now issue Warning #273:

"FILE SPECIFIED AS CHAINED ACCESS MUST BE RANDOM PROCESSED, ASSUMED R IN COLUMN 28."

12. SR #8137 The RPG Compiler has been enhanced to support the Calculation operation "TIME", as currently available on the IBM System/3 and the HP300. The TIME operation will return the current time or current time and date to the variable specified in the Result Field (columns 43-48). The Result Field Length (columns 49-51) must be specified as "6" or "12", and the Decimal Position Field (column 52) must be 0 (zero). The TIME operation is based on the 24 hour time system.

Factor 1, Factor 2, Half Adjust, and Resulting Indicator fields must all be blank for this operation.

The date returned to the Result Field may be foreign formatted if you put an I, J, or D in column 21 of the Header Card. A blank in this column indicates domestic format. See pages 3-3a to 3-5 of the RPG Reference Manual for descriptions of date formats and column 21 of the Header Card.

To direct the compiler to only return the current time in the Result Field, you put a "6" in the Result Field Length with "0" decimal positions. The format of the time digits returned is:

hhmmss

Where hh represents hours, mm represents minutes, and ss represents seconds.

To direct the compiler to return the current time and date in the Result Field, you put a "12" in the Result Field Length with "0" decimal positions. The format of the digits returned is:

If domestic format --> hhmmssMMDDYY
If foreign format --> hhmmssDDMMYY

Where MM represents the month of the year, DD represents the day of the month, and YY represents the year.

If a "6" or "12" is not specified in the Result Field Length, the compiler issues the following Error #675T:

"RESULT FIELD LENGTH IN COLUMNS 49-51 OF C CARD MUST BE 6 OR 12 FOR TIME OPERATION."

Likewise, if a "0" (zero) is not specified in the decimal position field, the compiler issues the following Error #676T:

"DECIMAL POSITION IN COLUMN 52 OF C CARD MUST BE 0 FOR THIS OPERATION."

13. The RPG Compiler has been further enhanced to support the new Calculation operation "TIME2". The "TIME2" operation will return to the alpha variable specified in the Result Field all or just portions of the following string formatted as such:

```
THU, JAN 10, 1980,  9:25 AM   JULIAN:010
```

With the TIME2 operation, you can extract just portions of the above string to obtain just the day, date, time, year, or anything you want in the string by specifying the starting location and length of the portion to be extracted.

To use TIME2, the Operation Field (columns 28-32) must specify "TIME2", Factor 2 (columns 33-42) must specify the starting position of the portion of the string to be extracted, the Result Field Length (columns 49-51) must specify the length of the portion of the string to be extracted, and the Result Field must specify the name of the alpha variable to recieve the extracted portion.

For example, to obtain the entire 40 character string, you put a "1" in Factor 2 and a "40" in the Result Field Length. (This directs the compiler to start at position 1 and extract 40 characters.) To extract just the time, you put a "20" in Factor 2 and an "8" in the Result Field Length.

Listed below are important locations and lengths required to extract common informative portions of the string.

```
THU, JAN 10, 1980,  9:25 AM   JULIAN:010
!    !       ! !  !      !          ! !
1    6       14 17 20     27        38 40
```

FACTOR 2 start loc	RESULT FLD LEN len of portion	PORTION RETURNED result variable
1	17	THU, JAN 10, 1980

1	27	THU, JAN 10, 1980, 9:25 AM
6	12	JAN 10, 1980
14	4	1980
20	8	9:25 AM
38	3	010

Factor 1, Decimal Position, Half-adjust, and Resulting Indicators fields should all be blank for this operation.

If Factor 2 is a literal constant then it must be an unsigned integer from "1" to "40" in columns 33-34. Columns 35-42 must be blank. If Factor 2 is not a valid constant in this range, the compiler issues the following Error #678T:

> "FACTOR 2 OR RESULT FIELD LENGTH MUST BE AN UNSIGNED INTEGER FROM 1 TO 40 FOR TIME2."

Similarly, the result field length must be an integer from 1 to 40.

If Factor 2 is a variable, then this variable must be defined with "0" (zero) decimals and with "1" to "4" digits. If the variable does not meet this condition, the compiler issues the following Error #677T:

> "DIGIT LENGTH FOR VARIABLE IN FACTOR 2 MUST BE 1 TO 4 DIGITS FOR TIME2 OPERATION."

When the starting location specified in Factor 2 is a literal constant, the compiler can determine if the portion length (specified in the Result Field Length) goes beyond the 40 character string. For example, if you specify "31" in Factor 2 and "20" in the Result Field Length, the compiler will issue the following Error #679T:

> "START POSITION SPECIFIED IN FACTOR 2 PLUS RESULT FIELD LENGTH IS BEYOND 40 FOR TIME2."

However, if Factor 2 is a variable, RPG cannot determine this bounds error until runtime when the variable actually stores a numerical value. If this error occurs at runtime, RPG will halt with Error #13: INVALID NUMERICAL DATA and expect a response from you. The runtime error will specify the program source line in error and the DB address of the variable in Factor 2. If you respond to this error with "0" (continue), nothing is returned to the Result Field variable.

B. CORRECTIVE SOFTWARE CHANGES

1. SR #7024 RPG now correctly merges a textfile with a masterfile into a newfile. Previously, if the masterfile was a null file, the first record of the textfile was not put to the newfile. Note also that $$Command records will be merged without sequence numbers in the listfile.

2. SR #6907 RPG did not perform final total-time operations (LR calculations and output) when the input file was one of the following:

   a. Equated to $NULL
   b. Contained zero records
   c. Had level breaks defined for some (not all) record types, but the file did not contain any records qualifying as one of those types. That is, a level break never occurred - not even on the first record - giving the file a "NULL" appearance with respect to total-time operations.

   Now, RPG will perform final total-time operations even if the input file is one of the above.

3. SR #7646 When an input file was being processed by an ADDROUT file, the Lookahead field was not set to 9's when the last record in the input file was read. At EOF of the input file, the compiler did not check to see if a Lookahead field was used. This has been fixed.

4. SR #7894 The MOVEL operation always moved the sign when moving data from a numeric field to a shorter or longer alpha field. MOVEL now handles the sign in a numeric to alpha move as follows:

   A positive zoned source always yields an unsigned result.

   ```
   Result shorter than source      -> zone not moved.
   Result same length as source    -> NEG. zone moved.
   Result longer than source       -> NEG. zone moved.
   ```

   This is the same way that IBM System 3 RPG handles the zone for numeric to alpha moves.

5. SR #7942 When using an 'E' in column 17 of the File Card for an Update Primary file which accesses an Update Secondary file using the FORCE calculation command, the program continued to process the remainder of the Secondary file after End-Of-File had been reached in the Primary file. The compiler was incorrectly treating this as a matching record situation. A flag has been added in the compiler to indicate that this is a FORCE access situation.

6. SR #8161 RPG failed with an unidentified record error when testing an input record on the digit portion on more than 1 column. Testing on one column ran correctly, but when testing more than one column, the compiler failed to reinitialize the X register before each digit test. This has been fixed.

7. SR #8327 Compile time Warning #269 was given in error. Now a file specification for a Display file with a device name of $STDLIST does not generate #269W.

8. SR #8427 LOKUP did not return the first best item when searching a table or array. The linear search did not save the first best item. And the binary search sometimes skipped the first best item because of the division by 2 used in the search technique. This has been fixed.

9. SR #9726 When using a 2608 Printer, a visible character NULL appeared at the end of the date heading. This null character has been overlayed with a blank character.

10. SR #9748 When using a USWITCH record to set User Indicators (U1 - U8) in a job stream, the switch setting did not work correctly if the same switch was first set on then off. This has been fixed.

11. SR #9749 When using a USWITCH record to set User Indicators (U1 - U8) in a job stream, the program would incorrectly abort with the message: "ERROR IN USER SWITCH INITIALIZATION, INVALID INPUT DATA" if the record was in free format (spaces between operands and operators). This has been fixed and the record may appear in free format.

    Example USWITCH: U1=ON, U1 = OFF , U3 = JCW

    Furthermore, the passing of JCW's from one job to another for User Switch initialization now works correctly.

12. SR #9803 When the user pre-selected error response 0 for division by zero (to continue program execution) and the division operands were arrays (the divisor array had some elements of zero value), the results produced were incorrect. Adjacent data on the stack may also have been affected. The runtime trap routine which handled the division by zero was not clearing the stack properly of all division operands and parameters. This has been fixed.

13. SR #9928 The indicator cross reference reported that an indicator which was Defined and Referenced on the same line was only Defined on that line. RPG now reports it correctly as Defined and Referenced on that line.

14. SR #9929 The Overflow indicator should have been set on when the overflow line was passed, not reached. This has been corrected, to be consistent with IBM System/3 usage.

15. SR #10158 All files (except IMAGE and SPECIAL files) were first closed with disposition = 2 (temporary job file rewound). If this failed (always the case for KSAM), the files were closed with disposition = 0 (no change). RPG now ensures that KSAM files are only closed once, with disposition = 0.

16. SR #10327 When First Record equals one was specified with a 1 in column 22 of the Header Card, RPG failed to retrieve the proper record from the KSAM file when the following conditions were all true:

    a. The KSAM file was initially built with First Record=0.
    b. The KSAM file was multiply defined with different access methods with DSNAME records.
    c. One of the multiple definitions of the KSAM file was processing by direct access and this definition was not the very first file description appearing for the KSAM file.

    Because DSNAME'd files are only opened once, RPG would only open the KSAM file at the appearance of the first KSAM file description, check First Record option, then proceed. When RPG processed the second file description (DSNAME'd to the same KSAM file), it determined that the KSAM file was already opened, and incorrectly skipped the check for First Record option. This has been fixed.

17. SR #6442 Error 553T (Control Level Length Differs From Previous Definition) was issued on the wrong control indicator. The immediate preceeding control indicator was the actual indicator in error. The compiler will now issue a new error message for this condition: Error 591T

    "THE PRECEDING CONTROL LEVEL LENGTH DIFFERS FROM ITS PREVIOUS DEFINITION."

    Note the added description "PRECEDING". The compiler will issue the error on the control indicator just after the actual indicator that is in error. This is neces-

sary for processing split control level length sums. The compiler cannot determine if level lengths for a control indicator are different until all definitions have been processed and lengths summed for that indicator.

When possible, the compiler will issue Error 553T on the control indicator in error; otherwise, the compiler will issue Error 591T on the control indicator just after the one in error.

18. SR #7872 When running RPG with V/3000 without a batch file and with a trace file, Error number 24 - Bounds Violation was incorrectly issued. This has been fixed.

19. SR #8410 When running RPG with V/3000, the field named NONDEL of Event 11 was always empty or non-existent. The compiler now correctly loads in the field.

    Also, the error message window was getting garbage if the error message duration was zero as specified in the File Specifications. This has been corrected.

20. SR #8411 When running RPG with V/3000, the compiler incorrectly truncated the form name to 8 bytes after execution of the "RESUME" instruction to return to collect mode from browse mode. This has been fixed.

A. ENHANCEMENTS

1. [ ]VI System Function

   This function takes as its right argument a character vector and returns a boolean vector as its explicit result. The length of the result is equal to the number of non-blank character groups in the argument. For each group of non-blanks, the corresponding element of the result is set to 1 if and only if the non-blank characters form a valid APL numeric constant.

2. [ ]FI System Function

   [ ]FI takes the same argument as [ ]VI, returns a result of the same length as that of [ ]VI, but the result is numeric. For each group of non-blank characters in the argument, the corresponding element of the result is set to 0 if the group does not form a valid APL numeric constant. Otherwise, it is set to the value of the constant itself.

3. [ ]PRINT System Function

   The [ ]PRINT system function provides a quick and easy way to send APL data to the line printer or to a disc file from an APL session or function. [ ]PRINT takes as its right argument any APL variable or expression that has a value. The optional left argument is a character vector representing the termtype of the receiving file or device. The default value is 'ASCII'. [ ]PRINT takes the right argument, formats it in the same manner as monadic format, and writes the result to a file whose formal designator is APLLP. This creates a spool file that is printed immediately. Suppose that you wanted each successive call to [ ]PRINT to append its results to a disc file. The following MPE commands, entered before invoking APL, or during the APL session using the)MPE system command or shared variables, would produce the desired results:

   ```
   :BUILD filename;REC=recsize,blockfactor,F,ASCII
   :FILE APLLP=filename,OLD;DEV=DISC;ACC=APPEND
   ```

   It is important to note that [ ]PRINT will not return an error message even if you are attempting to append records past the end of the disc file. Before invoking [ ]PRINT, it is advisable to set the value of [ ]PW (print width) to the record size of the APLLP file.

4. [ ]CSLOC System Function

   The left argument of [ ]CSLOC is a character scalar or vector. The right argument is a character scalar, vector, or matrix. If the right argument is a scalar or vector, then the result of [ ]CSLOC is an integer vector containing the starting indices of each occurrence of the left argument in the right. If the right argument is a matrix,

then the result is an integer vector containing the row numbers in which the left argument appears, left-justified, in the right. In either case, an empty vector is returned if there are no occurrences of the left argument in the right. []CSLOC is particularly useful in text processing and table searching.

5. []CSMOD System Function

   The character string modifier system function takes as its right argument a character vector to be modified. The left argument is a character vector of the form:

   <delimiter> <string1> <delimiter> <string2> <delimiter>.

   Execution of []CSMOD causes all occurrences of string1 in the right argument to be replaced by string2. The modified character vector is returned as the result. String2 may be of length 0. This would, in effect, delete all occurrences of string1 in the right argument.

6. []CSD System Function

   The character string delimiter system function takes a character vector as its right argument and a character scalar as its left. The explicit result is a character matrix formed by scanning the right argument (left-to-right) and starting a new row of the result at each occurrence of the left argument (the delimiter). Consecutive, leading or trailing delimiters cause empty rows to be formed. The delimiter characters do not appear in the result. If there are no occurrences of the delimiter in the right argument, then a one-row matrix is returned.

7. []APPENDR System Function

   This function causes the same action as []APPEND. The difference is that []APPENDR returns an integer scalar specifying the number of the component into which the data was placed.

8.)* System Command

   This command speeds up the debugging of APL programs. Its invocation cause the APL editor to be entered with the function currently on top of the state indicator stack. The line containing the last error is displayed and the editor is placed in modify mode.

9.)SHOW System Command

   This command allows the user to see how variables are stored in the workspace. The command is optionally followed by a list of variable names. For each name, a display of the symbol table information for that variable is produced. If no list of names follows the)SHOW command, then all the variables in the workspace are displayed.

10. Editor Enhancements

a. Execute Command
If the first character in an editor command line is an execute character or a right bracket (]), then the rest of the line is executed as though it had been entered in calculator mode (excluding system commands).

b. Put Command
The put command, specified by a left arrow, adds the rest of the line to the edit text after the current cursor position.

c. Fetch Command
Specified by a right arrow, the fetch command takes as its argument a single line number. It copies that line of the edit text into the command buffer with the execute symbol inserted in front, allows the user to edit it in modify mode, and then interprets the line as an editor command string.

11. Change to]VR and]CR
These two system functions now take an optional left argument whose value is boolean. If this value is 1, then the result will contain bracketed line numbers on each line. If the left argument is zero or if it is omitted, then the system function will behave as it did previously. Note: When using a workspace created with a prior version of APL\3000, a syntax error may occur when trying to use]CR or]VR dyadically. This is because the workspace's symbol table still believes these functions to be only monadic.)COPYing the workspace will resolve this problem.

B. CORRECTIVE SOFTWARE CHANGES

1. [SR #10866] No check was being made to see whether the left argument of a reshape was negative if the right argument was a matrix and the left was a scalar. This should have produced a DOMAIN ERROR. This problem has been corrected.

2. [SR #9571] Incorrect answers occurred when applying the "take" operator to a matrix in a function when on first execution, the result was smaller than the right argument (undertake), and on subsequent executions the left argument was such that the result of the "take" was larger than the right argument. This has been fixed.

3. [SR #7826] An excessive number of binding errors were occurring on expressions containing subscripted assignments. The problem was intermittent and only affected performance, not the validity of the results. The problem has been corrected.

4. [SR #9836] Using the niladic right arrow to clear the state indicator stack sometimes caused a system error. This no longer happens.

5. [SR #10609] When trying to tie an APL data file, a FILE ACCESS ERROR occurred if the access matrix of the file contained a row whose elements were 0 and -1 if the creator of the file used a high minus instead of a negative symbol. This has been corrected. Remember that an APL data file and its associated key file have to be released with MPE before a different user can tie the file.

6. [SR #10714] A problem in converting data types during the execution of catenate was causing a system error in very rare cases. This problem has been fixed.

C. MISCELLANEOUS

1. The default value for]VM (virtual memory) has been changed so that, in a clear workspace, there are now 24 pages of 1024 bytes each. This was done to take advantage of the increase in the average memory size of installed systems.

2. Several significant performance improvements have been made to certain operators including dyadic epsilon and dyadic iota. However, the benefits of these optimizations will not be realized unless functions containing these operations are recompiled. This can be done most simply by entering the APL editor with the function and then exiting the editor with the END command.

A. ENHANCEMENTS

1. The operator console command =DSLINE has been replaced by the Unified Command Language (UNCL) command :DSCONTROL, which had been under separate documentation.

2. DS error 214 has been modified so that it is only called when a :DSLINE command has not been performed. DS error 227 has been added to inform the user that a required :REMOTE HELLO has not been done.

B. CORRECTIVE SOFTWARE CHANGES

1. SR #9953 :REMOTE REMOTE BYE previously lost the # prompt character from the intermediate system and left the user hung, requiring a logoff from the local session. This has been corrected.

2. SR #10600 An =ABORTJOB of a job/session that had a pending :DSLINE open request (or an =ABORTIO on the DS line under the same circumstance) caused an internal table to record a 'software abort' error status for the job/session process number. Any subsequent job/session with the same process number attempting to use that DS line would get the 'software abort' error, and would be unable to use the line. This problem has been corrected.

3. SR #11014 An open failure on the communications line for a :DSLINE.....;EXCLUSIVE open request no longer leaves the line in inaccessible exclusive mode.

4. SR #11275 A PTOP slave program which terminates without calling GET will now return an error code of 226 to the POPEN intrinsic, rather than the spurious error 70 that was returned.

5. SR #11303 The POPEN intrinsic will now accept a lower case DS device name.

A. ENHANCEMENTS

The MPLINE command is now a user command, not an operator command. Users may be able to use the command by being ALLOWed by the operator or by being ASSOCIATEd with the deviceclass for the line (usually MPSUPER). So now the MPLINE command may be used in UDC's or programmatically.

B. CORRECTIVE SOFTWARE CHANGES

The following problems have been fixed.

1. If the configuration file name in the MPLINE command was larger than eight characters, a system failure occurred.

2. If the logical device number entered was larger than the maximum line number in the system, a system failure could occur.

3. There was a potential debug call in the Multipoint Monitor.

4. The data set ready delay parameter in the configuration file was not checked correctly.

5. Two error messages (invalid multipoint terminal subtype or subtype incompatibility) were printed with a logical device number of zero for the line number.

A. ENHANCEMENTS

1. SR #7625 - Tradeoff number for disc space versus I/O efficiency has been raised from 200 to 400 records in the file. Any file with <= 400 records will be optimized for disc space and any file with > 400 records will be optimized for I/O efficiency in the EDITOR.

2. The KEEP file is now opened with NOBUF access so a complete block of data is written at a time. This speeds up the KEEP.

B. CORRECTIVE SOFTWARE CHANGES

1. SR #7435 - It is no longer possible to KEEP a file that has a record length too large to be texted back into EDIT/3000. Now, if a record length is greater than 248 bytes, it must be kept unnumbered.

2. SR #7845 - The file system error is now displayed when a "*25* SCRATCH FILE OPEN FAILURE" occurs while TEXTing in a very large file.

3. SR #7930 - EDITOR will no longer override the blocking factor specified in the :FILE equation for files on non-disc devices.

4. SR #8022 - When changing a "string" to "" (null), only the occurrence of the string at which the cursor is pointing will be changed rather than all contiguous occurrences of that "string".

5. SR #8067 - EDITOR will now allow lower case "offline" in XPLAIN command.

6. SR #9156 - When TEXTing in a KEEPQ file, the VARIABLE/FIXED attribute is now set to whichever was in effect when the file was kept. Note: All quick files which were kept under previous versions will set the FIXED attribute; if they are intended to be VARIABLE, you should do a SET VARIABLE after TEXTing them in the first time with this version.

7. SR #9196 - CHANGE "" to "string" can no longer be done beyond valid data on a line.

8. SR #9372 - The command sequence:

```
/DELETE ALL
/LIST ALL
/DELETE
/DELETE
```

previously caused garbage line numbers to be printed and would give error #15 when trying to ADD. You will now get error "*40* UNDEFINED TEXT" when trying to do the DELETE following the LIST ALL and the ADD will work correctly.

9. SR #9434 - If a GETDSEG fails while TEXTing in a file, further attempts to TEXT the file will no longer result in error "*66* FREEDSEG REQUEST DENIED."

10. SR #9693 - If the left margin is moved towards the right and then a F "string" is done, the search will now begin at the new left margin even if it is to the right of the cursor position.

11. SR #9964 - When a command line > 256 characters is entered in a session using "&", the EDITOR will no longer abort. It will give error "*46* COMMAND CONTINUES PAST 256 CHARACTERS" and allow the user to reenter the command line.

12. SR #10009 - The double slash, "//", may again be used to reset Z:: in a USE file.

13. SR #10428 - A file can now be kept when a temporary file of the same name exists.

14. SR #10667 - EDITOR will now handle line numbers correctly when joining a COBOL file to a default format file and vice versa.

15. SR #10668 - When a range is specified in which the start and end line numbers both precede the first line number in the file, error "*39* RANGE IS NULL OF FIRST POSITION FOLLOWS SECOND" will be returned instead of "*44* LINE NUMBER ZERO CAN NOT BE ACCESSED."

16. SR #10669 - CHANGEQ now works correctly when the string being changed has zero characters. It used to print the line even though the Q was specified.

17. GATHER will no longer attempt to add more lines to the work file than there is room for. You will now get error "*36* SCRATCH FILE IS FULL. KEEP, THEN TEXT AGAIN" instead of "*31* SCRATCH FILE WRITE ERROR."

18. When attempting to TEXT in a file that does not exist, a tombstone is printed and the previous work file is still available for editing. However, if that file was unnumbered, a DMOVIN ERROR would occur whenever a line number was referenced. This error will no longer occur.

C. DOCUMENTATION CHANGES

1. Page 3-60a: Under "limitations", add the following paragraph:

   Files with a fixed record length greater than 256 bytes may not be texted in. Attempts to do so will result in error #24, RECORD SIZE TOO LARGE.

2. Page 3-36b: Add the following paragraph:

   EDIT/3000 will not allow keeping numbered files which, when sequence numbers are included, will have a record size greater than 256 bytes. This will prevent files from being kept which can not be texted in later. Attempts to keep such a file will result in error #75, RECORDS TOO WIDE - KEEP UNNUMBERED.

   Also add to page 5-6:

   ERROR NO. 75
   TYPE: SOFT
   MESSAGE: RECORDS TOO WIDE - KEEP UNNUMBERED
   CAUSE: Record size of KEEP file is greater than 256 bytes.
   CORRECTIVE ACTION: KEEP the file UNNUMBERED or SET LENGTH to a value less than or equal to 248.

   (Remember that before shortening the length it may be necessary to shorten the right margin.)

A. CORRECTIVE SOFTWARE CHANGES

FORMAINT

1. [SR #3898] Running FORMAINT without save files capability and attempting to create a form resulted in the system either running out of virtual memory or disc space.

   This condition has been corrected. FORMAINT displays a message informing the user that he does not have save file capability, rings the bell and returns to the main menu.

2. [SR #4882] On a 2640B terminal Block/Line mode worked, but in Block/Page mode the program would autoread.

   Now, both modes will work correctly.

3. [SR #4937] Pressing function keys F1-F7 caused FORMAINT to report an unrecoverable error. If this was done when a form was being modified or created, then the form was lost.

   This condition has been corrected.

4. [SR #5385] While modifying an existing form, and running out of room in the form file, FORMAINT did not allow the user to return to the main menu.

   This condition has been corrected. The user can now return to the main menu by pressing the F8 function key.

5. [SR #5566] FORMAINT would destroy forms in a forms file if it encountered an end of file during a modify operation.

   This condition has been corrected.
   FORMAINT will display the following messages.
   FORM NOT UPDATED!!
   NO ROOM IN FILE FOR MODIFY (PRESS F8 KEY TO RETURN)

6. [SR #6341] FORMAINT would not let the user exit the modify menu by using the F8 function key.

   This condition has been corrected. The user can now exit the modify menu by using the F8 function key.

7. [SR #7109] FORMAINT required the D & G straps to be closed on a 2640B terminal and only allowed Block/Line mode.

   The FORMAINT program has been modified to allow Block/Page mode operation on a 2640B terminal.

8. [SR #7454] When an invalid form name was entered in the modify menu, FORMAINT would hang up and not allow the user to exit except by aborting.

   This condition has been corrected. The user can now exit by pressing the F8 function key.

DEL PROCEDURES

9. [SR #5978] Page mode read with Readterm occasionally gets -2003 errors.

   The DEL routine which strips out US/RS chars begins one byte before the buffer. If that byte contained the bit pattern for a US or RS, a -2003 error was returned. This condition has been corrected.

10. [SR #6885] In version 1.07 CLOSETERM truncated the form on the screen when it was greater than 24 lines.

    This condition has been corrected. CLOSETERM has been modified to position the cursor past the last line of the form on the screen.

11. [SR #7797] There were random but consistent -2003 errors on any MTS terminal from CREADTERM.

    This condition has been corrected. (See solution for SR #5978.)

B. MISCELLANEOUS

1. [SR #4819] Hitting the break key or coming to any abnormal termination while creating or modifying forms with FORMAINT will corrupt the forms file and may result in a bad end of file pointer and/or over-written forms.

   Hitting the break key during a page read will cause the terminal to stop transmission.

2. [SR #4966] Problem running Page mode DEL over DS line.

   When opening the DS/3000 communication line the DS/3000 line buffer size must be set to the maximum amount of data expected to be transmitted by the user°s terminal in Block mode at any one time.

3. [SR #5488] User sending ESC "E" immediately before calling COPENTERM on a Full Duplex 1200 Baud line will loop.

   Sending an ESC "E" to a 264x terminal causes a full reset of the terminal. The user must wait up to a few seconds before sending additional data. See the reference manuals for the various 264x terminals and exact wait times.

4. [SR #4067] DEL/3000 procedures fail to work properly on a Series I if the term type equals four.

   On a series I the user should logon with TERM=10, since this cannot be altered programmatically on the Series I. On the Series II and III, the MPE TERMTYPE is set to 10 by OPENTERM if the logon TERMTYPE was not 10, or if the configuration default is not 10. The logon TERMTYPE is restored when CLOSETERM succesfully executes.
   See DEL reference manual page 2-7.

KSAM/3000 HP32208A.03.01

NOON208A.HP32208.SUPPORT

A. CORRECTIVE SOFTWARE CHANGES

1. SR #10467 KSAMUTIL required a file name for the SAVE command. The HELP command showed the file name as optional, which, if not specified, caused error message 1006 (FILE NAME TOO LONG OR ABSENT) to be displayed. This has been fixed.

2. SR #8083 KSAMUTIL did not do any file name checking for special characters. This caused other files to be possibly purged or saved. This has been fixed.

3. SR #11390 When KSAM created a new KSAM file with eleven or more keys, System Failure #314 (PROCESS ABORTING WITH SIR) occurred. This has been fixed.

A. ENHANCEMENTS

1. [SR # 7120]
A crash protection feature for batch files has been incorporated to limit loss of data in the event of a system crash.

A new intrinsic VPOSTBATCH has been provided, which:
a. posts an end of file mark after the last record referenced.
b. updates the environmental information into the batch file labels.
In the event of a system crash, all data up to this point would be preserved, and collection of data would resume from this point.

In ENTRY, the posting interval is set to 20. The user could however change it(variable PARMVAL = interval).
NOTE: COM'NUMRECS may not be restored correctly after a crash.

2. [SR # 7059]
When a negative value was entered into a field with processing specifications "FILL LEADING '0'" or "STRIP LEADING '0'", the filling/striping was done to the left of the sign.

The statments STRIP/FILL LEADING are now handled slightly differently by the intrinsics, for negative numeric fields(IMP,NUM,DIG,IMPn,NUMn).
a. FILL LEADING "char"
For a negative numeric field, if char is numeric, the filling is done to the right of the sign, else it is done to the left of the sign.
b. STRIP LEADING " chars"
For a negative numeric field, stripping is done both to the left and right of sign (the sign is also stripped if it is a strip character).

3. To improve V/3000 performance, the nature of the forms/reformat files have been substantially changed.

The forms/reformat files are no longer KSAM files. To ensure compatibility, any existing V/3000 KSAM files will still operate successfully, but any fast/slow forms files created with this version will be a non-KSAM file. BASIC users and anyone else who defines a COMAREA extension must increase their extension length by 500 words for fast forms files, and 1300 words for non KSAM

slow forms files.

The new files are 256 byte fixed length ascii files.The keys and forms records are interspersed together in the file.

The advantages of the new format are:

* Improved crash resiliency: In the event of a system crash, in almost all instances, only the form being worked upon during the crash, might be damaged. In most instances the user can recover the form by reentering all the menus of the form (for the field menu, hit toggle before pressing enter).

* Improved run time performance: The access method for the new files does not use extra data segments for the key blocks. This substantially reduces system memory requirements for a V/3000 application. Because the access method is smaller and more specialized, it is faster. However, the access method requires 500 words (1300 words for slow forms files) of stack space to store the key blocks.

4. INTRINSICS have now been provided to obtain information about a forms file, a form, or a field in a form.

   a. The INTRINSIC VGETFILEINFO obtains version,number of forms, maximum data buffer length, number of save fields, head form name, error enhancement, window enhancement, window line.
   b. The INTRINSIC VGETFORMINFO obtains form name, form number, number of fields, data buffer length, next form name, repeat option, freeze option.
   c. The INTRINSIC VGETFIELDINFO obtains field name, screen order and number, length, data buffer position, enhancement, data type, field type.

5. To improve V/3000 performance for data base applications, the concept of a family of forms has been introduced. The family is called a multi-usage form family. All members of the family share the same screen definition, but can have different field characteristics, or field processing specifications, and repeat, freeze and next form options.

   The root member of the family is called a parent form, and all the other members are called son forms. The screen definion of the sons cannot be changed. Changing the screen of the parent changes the screen definition of the whole family. Deleting the parent deletes the whole family.

A son is reproduced from a parent, or any other member of the family. Reproduction causes the son to inherit the field characteristics of the member from which it is generated.

The concept of form families allows optimizations of terminal I/O between members of the same family. The aim of the concept is to enable the data entry person to apparently use the same "form" repeatedly a number of times but have the fields behave differently during each usage.

6. To reduce compilation time for large forms files, an incremental compilation feature has been provided. When the forms file is compiled, only the forms changed since the last compilation are compiled. The compilation step for the rest of the forms has been reduced to simple housekeeping functions. However if a save field is added/ modified/deleted, all the forms are compiled.

7. The FCONTROL 30/31 feature implemented in MPE is used by version A.01.01 of V/3000. This delegates terminal protocol handling down to the driver level, thus slightly improving the response time, and reducing the instances of terminal hangs, in multi process applications (SR#9949).

A. CORRECTIVE SOFTWARE CHANGES

1. [SR # 6106]
In FORMSPEC, the keyboard wasn't locked while displaying parts of the FIELD MENU (specifically, the processing specifications and the lines of the screen that contain the field being described). This has been corrected.

2. [SR # 6107]
If the CLOSE/OPENTERM intrinsics were called while the stack was less than 176 (%260) words, an abort with a bounds violation occured. This has now been corrected.

3. [SR # 6329]
If the form definition contained a shift out followed by "[", FORMSPEC incorrectly interpreted it as the start of a field. This was because FORMSPEC interpreted all shift out characters as regular characters. FORMSPEC now distinguishes between regular and shift characters, and treats all shiftout characters as text.

4. [SR # 6439]
FORMSPEC now sets the default error enhancement to IU as per the manual(instead of IUHB).

5. [SR # 6647]
COBOL/ENTRY did not allow the use of an outdated batch file. This has been corrected.

6. [SR # 6767]
If an invalid entry was keyed into the ENH field of a field menu, FORMSPEC allowed the form to compile even though the invalid entry is still displayed in the field menu. FORMSPEC now displays the default enhancement in the field menu.

7. [SR # 6820]
COBOL/ENTRY contained a superfluous para DISPLAY-MSG. This paragraph has been deleted.

8. [SR # 6839]
FORTRAN/ENTRY did not close the forms file if the user did not want to continue with an outdated batch file. This has been corrected.

9. [SR # 6840]
V/3000 did not maintain save field values for a batch file between sessions.
This has been corrected.

10. [SR # 6842]
Strip all statement did not strip the last character in a field if it was a strip character. This has been corrected.

11. [SR # 6948]
Line 3210 in the BASIC/ENTRY now reads ---
E$= "DELETE key defined only for BROWSE".

12. [SR # 7037]
The main menu was not cleared after each use. This has been fixed.

13. [SR # 7060]
Setting a num field to 0 in the processing specs caused an invalid value to be stored. This has been corrected.

14. [SR # 7129]
Invalid field names over 15 characters long in the field processing specifications were undetected, and caused FORMSPEC to loop during compilation. FORMSPEC now detects invalid identifiers in the field specifications.

15. [SR # 7210]
In FORMSPEC, the fast forms file compile option sometimes generated errors 172 and 190.
This has been corrected.

16. [SR # 7244]
If two concurrent users used ENTRY to entered data into identically named new batch files, one of them would lose his batch file, on termination. The intrinsic VOPENBATCH has been modified to make new batch files, permanent files.

17. [SR # 7406]
The intrinsic VCLOSEFORMF did not release the DL/DB area allocated by VOPENFORMF. The intrinsic now releases the DL-DB area. Applications using more than one forms file must close them in the reverse order of opening.

18. [SR # 7466]
The last character of an IMPn (n > 0) field was not checked for validity. This has been corrected.

19. [SR # 7632]
In FORMSPEC, if a field's length was changed on the screen and the field menu for the screen was not "entered", the old length of the field was displayed when the form was listed. This has been fixed.

20. [SR # 7819]
The processing statement IN $STATE aborted if the field contained all blanks. This has been corrected.

21. [SR # 7873]
If a user exited from Browse mode, ENTRY resumed data collection from the wrong form. This has been fixed.

22. [SR # 7915]
FORMSPEC sometimes aborted with a dynamic array error, when adding or copying a new form. This has been fixed.

23. [SR # 7941]
Blank numeric fields were flagged as errors in reformat. REFORMAT now assigns a default value of 0 to blank numeric fields.

24. [SR # 8026]
VREADBATCH died not set dbuflen as documented. This has been corrected.

25. [SR # 8062]
FORMSPEC returned obscure messages when the forms file was full. The error messages have been changed to suggest that the user check if the file is full.

26. [SR # 8129]
FORMSPEC aborted with a bounds violation when listing if the last line of a processing statement (not all blanks) ended in a blank. This has been fixed.

27. [SR # 8419]
REFSPEC aborted with a dynamic array error during compilation if a number of reformatting specifications were entered for an output field. This has been fixed.

28. [SR # 9159]
V/3000 did not always recognise a 2647 terminal. This has been corrected.

29. [SR # 9320]
Comparison of equal quantities in the field processing specs sometimes failed. ex.. "IF 0.2 + 1.0 + 1.0 NE 2.2 THEN FAIL" did not perform correctly. This occurred because V/3000 performs all numeric operations with long real operands, and roundoffs during conversion caused problems. This has now been fixed.

30. [SR # 9383]
V/3000 did not always detect a KSAM write error resulting in obscure error messages during reads. The detection of KSAM write errors has been substantially improved.

31. [SR # 10173]
The BASIC version of ENTRY did not detect edit errors in browse mode. Now it does.

32. [SR # 10396]
The V/3000 run time intrinsics did not operate correctly with a BASIC program (interpreter) using INVOKE. This has been fixed.

33. [SR # 10544]
A call to VWRITEBATCH without opening the batch file resulted in a bounds violation. The intrinsic now detects and reports this error.

34. [SR # 10588]
The textual discription "TERMINAL or FORMSFILE not yet opened" was missing for runtime intrinsic error # 45. This has been fixed.

35. [SR # 10758]
If the window was on line one, and the form on the screen had no unprotected fields, appending a form sometimes caused the form to be overlayed over parts of the previous forms. This has now been corrected.

36. [SR # 10771]
The statement STRIP ALL " " did not strip blanks. This has now been corrected.

37. [SR # 10837]
VSHOWFORM did not write to fields in lines > 99. This has been fixed.

B. DOCUMENTATION CHANGES

1. The intrinsic VOPENTERM performs a reset terminal function. If a tape cassette is not at the load point, this causes the cassette to rewind, and can cause VOPENTERM to fail due to a status request read timeout. (Remove or put cassette at load point before running ENTRY, FORMSPEC, REFSPEC, or any program that uses VOPENTERM.)

2. In FORMSPEC, if format mode is turned on manually when designing or modifying a screen, format mode must be manually turned off before ENTER is pressed.

3. When using FORMSPEC or REFSPEC, if lockwords are present on forms files, reformat files, or key files, the lockword must be given with the file name, unless a file equation containing the lockword was given before running FORMSPEC or REFSPEC. This is due to the fact that FORMSPEC and REFSPEC prompt for the file names in block mode with format mode on and the system will automatically prompt for the lockword(s) expecting a character mode terminal.

4. When using FORMSPEC, REFSPEC, ENTRY, or the intrinsics VREADFIELDS or VSHOWFORM, there must be sufficient terminal buffers available for all concurrently executing terminal I/O operations. It is recommended that the number of terminal buffers be at least 150. Creating, modifying, or displaying a form of 4000 characters requires 134 terminal buffers. Terminal buffers may be set to a maximum of 255, shared by all processes. (See configuration dialogue in System Manager manual.)

5. When using a remote terminal via the DS facility, some forms over 255 characters long are not correctly displayed. This occurs when the LINEBUF parameter is not used in the DSLINE command. When forms larger than 255 characters are being used, include LINEBUF=n in the DSLINE command, where "n" is the number of words required by the screen image of the largest form in the forms file.

6. In the intrinsics VINITFORM, VFIELDEDITS, and VFINISHFORM, because all leading and trailing blanks are stripped from a field before a MATCH statement is executed, any pattern that requires leading or trailing blanks will always fail.

7. If some of the parts of COMAREA that should be initially set to zero have non-zero initial values, the results of some intrinsics are unpredictable. A program must insure that the initialization of COMAREA is done before the first call to the first V/3000 intrinsic invoked. The values in COMAREA should not be changed between calls to V/3000 intrinsics except under documented conditions.

8. The contributed utility program RESTORE (:RUN RESTORE...) will not correctly copy records over 2000 bytes. If a KSAM forms file or a reformat file is copied from a store STOREto a disc file using this program, the results are unpredictable. (Use the command :RESTORE and FCOPY.)

9. Field language run-time error handling:

   a. An edit statement failure causes the current field to be flagged in "ERROR" and processing to stop for the current field.

   b. Any run-time processing failure (e. g. divide by zero; illegal indexed retrieve statement; etc.) causes the current field to be flagged in "ERROR" and processing to stop for the current field.

   c. If any field used in a statement is in "ERROR", the processing simply stops for the current field.

10. The beginning delimiter of a field ([) can be used as the terminating delimiter for the previous field.

11. Either the displaying([,]) or nondisplaying(esc[,esc]) field delimiters may be used. However for mixed useage, use the delimiters specified:

```
Instead of [,  esc]   use -->      [esc[, esc]
                             or    [ cntl F2, cntl F3
Instead of esc[,  ]   use -->      esc[,  esc]]
                             or    cntl F2, cntl F3 ]
```

12. If a system crash occurs while a form or reformat is being defined/modified, the following actions recover the file under most situations. (this is restricted only to KSAMles forms files.)

    * When the crashed file is next specified for modification in formspec or refspec, the access method automatically recovers all records in the file.
    * To ensure mutual consistancy of all records defining a form, reenter all menus of the form/reformat being worked upon at the time of the crash (for field menus hit toggle before pressing enter).

13. The contributed utility program CONCAT(:RUN CONCAT..) allows concatenation of two KSAMless fast forms files. This allows creation of fast forms files with considerably more forms than the current V/3000 restriction of 255.

COMPILER LIBRARY/3000 HP32211D.00.10

NOON211D.HP32211.SUPPORT

The Compiler Library source will be resequenced and combined with the maintenance file using this maintenance file (NOON211D). This will be done by using the control card $EDIT SEQNUM=1000 when compiling. The new file that results will be the resequenced source called D.01.00 and released on the next MIT.

CORRECTIVE SOFTWARE CHANGES

1. SR# 9837 - Unformatted (binary) READs or WRITEs of odd length character string arrays were performed inconsistently. The possibility of incompatible files has been eliminated, except as described in the Software Status Bulletin for this service request.

2. SR# 4210 - The tabulate edit descriptor (Tn) caused the rest of an output line to be lost when it was the last edit descriptor in a FORMAT statement used. This problem has been corrected.

3. SR# 4596 - The formatter only checked the first element of a character array for a string mismatch with the FORMAT statement. The formatter now checks every element of a character array.

4. SR# 7127 - The PRINT'FILE'INFO intrinsic reported the extent size as a negative value for extent sizes greater than 32767 This problem has been corrected.

5. SMR #7557 - When using a user-defined trap procedure for INTEGER*4 OVERFLOW, if the offending machine instruction was a stack op, control may not have been passed correctly to the procedure. A BOUNDS VIOLATION or INTEGER*2 OVERFLOW error could result. This problem has been corrected.

6. SR# 9859 - The position of the question mark prompt for an ACCEPT statement relative to the last character output on $STDLIST was unpredictable. The prompt now appears on the same line next to the last character output on $STDLIST.

FCOPY/3000 HP32212A.3.11

NOON212A.HP32212.SUPPORT

A. ENHANCEMENTS

1. SR #8122 - The KEY= option can now be used with the NEW option to build a new KSAM or MPE file. Previous versions of FCOPY would return error "*57* SYNTAX ERROR: ILLEGAL COMBINATION OF OPTIONS."

2. SR #9053 - The FILES= and the FROM=*/TO=* options can now be used for cartridges. Their use should be equivalent to use on magnetic tapes. This was accomplished through various combinations of Find file and Skip lines.

3. SR #7811, 9158, 9321, and 9849 - The WARN bit in the JCW is now set in a batch job by any error or warning message that does not set the FATAL bit. The FATAL bit is now also set by errors:
   134 FOUND EOF IN TOFILE
   302 RAN OUT OF VERIFY ERRORS
   303 RAN OUT OF IGNERRS
   304 RAN OUT OF COMPARE ERRORS

4. SR #9508 - The tofile is now opened write (save) only. This prevents the file from being deleted even though the user decides not to continue the copy after getting warning *200* or *201*.

B. CORRECTIVE SOFTWARE CHANGES

1. The record size for $HARD will now be taken from the FILE equation for HARD.

2. SR #8068 - The SAME enhancement now guarantees that if the first line says: SAME TO xxxxxx-1, then that line either contains all blanks or all zeros depending on whether ASCII/BINARY.

3. SR #8107 - ASCII reads on multipoint terminals now work correctly. In version A.3.08 bounds violations or other errors might occur. In version A.3.09 the EOF would not be recognized and caused blank records to fill the TO file. Also, when copying from cartridge to cartridge (an inappropriate use of FCOPY), FCOPY would hang on a BINARY write of a record that only contained CR-LF because the CR-LF was being stripped. The CR-LF will no longer be stripped in BINARY mode.

4. SR #9082 - If a VARIABLE length file is sent to the terminal and a record contains only 1 space a garbage byte is no longer added. FCOPY would previously print two characters, the blank and the second character of the previous line. Also, the dump option now works with a line of length zero.

5. A few cases of echo not being turned on and off for cartridges have been fixed.

6. If the FROM file is a cartridge, then the record format will be set to F instead of U.

7. SR #9662 - The TITLE is no longer printed at the foot of the page by a printer which is configured as a terminal.

8. SR #10058 - An extra byte is no longer added to records by the CCTL option when copying from a file without a carriage control to a file that has a carriage control.

9. SR #10478 - FCOPY will now always save a NEW file even if the preceding copy was ended by an error. It was previously purged when copying from $STDIN and ending the input with :EOD.

10. SR #10670 - The message "IGNORED ERROR x: READ ERROR FROM FROMFILE AT RECORD x" now prints the correct record number. Also, error "*303* IGNERR OPTION RAN OUT OF IGNERRS AT FROMFILE RECORD x" is now printed only when the specified number of errors is exceeded.

11. SR #10672 - Specifying TO=* in the first copy command after entering FCOPY now results in a syntax error instead of error "*109* ERROR IN CALLING FGETINFO FOR TOFILE."

12. SR #11123 - A garbage byte is no longer added to the beginning of records when copying from cartridge to disc at low speeds on a Series 33.

13. A garbage byte was added to the end of odd byte records when copying to a KSAM file from a file with undefined length. That byte is now blanked out by FCOPY. See KSAM SR #10901.

C. DOCUMENTATION CHANGES

1. Page 2-24: Change the first sentence to:
   The FILES function of FCOPY allows the user to copy multiple files from both labeled and unlabeled magnetic tapes and from cartridge tapes.

2. Page 4-8: The word "WARNING" should be deleted from message #134. This is now an error.

3. Page A-1: Delete the sentence in the third paragraph concerning the "*" notation.

COBOL C HP32213C.02.05
COBOL B HP32213B.03.05

NOON213C.HP32213.SUPPORT
NOON213B.HP32213.SUPPORT

A. ENHANCEMENTS

1. Code generation involving address constants has been improved.

2. Word moves will be used instead of byte moves if the source, target and length are even. Also one byte, one word, and double words are specially treated. The word moves are 4 times as fast as byte moves.

3. When a :SETDUMP command is issued and an error occurs in a COBOL runtime routine, DEBUG will now be called.

4. If the file in the GIVING clause of the SORT verb is NEW, then it will be closed TEMP (SR # 5689).

5. If a MOVE needs blank filling of exactly one blank, then a zero length move will no longer be generated.

6. A few other minor code generation enhancements were done. The extra load and delete at the end of non-PERFORMed paragraphs will no longer be generated. If error 403 occurs (EOF on symbol table files), then the paragraph and section will be printed.

7. Improved code generation for the VALUE clause for dynamic subprograms can reduce the initialization code segment by up to 10%.

8. If a VALUE clause is shorter than the PICTURE, up to 12 blanks will be added to the string, saving USL header space and permitting more value clauses.

9. Carriage return can now be included in non-numeric literals.

10. If the COBLIST is a disc file that does not exist, then it will be built. A file equation should be used to insure efficient space allocation of the listfile (SR #9799).

B. CORRECTIVE SOFTWARE CHANGES

1. A one word COMP item with right P-factor will no longer cause the compiler to abort on a DIVIDE.

2. If NOLIST is in effect then CODE will not be listed.

3. SR #3757, #7248, #7939 - If a DIVIDE needs an alignment such that the intermediate result is more than 28 digits, then the shift will be reduced at the cost of decimal places in the result.

4. SR #4032 - Non-integer COMP compares against an index will now be flagged.

5. SR #7062 - A REDEFINE can now appear under a REDEFINEd 01 level in the LINKAGE section without an error.

6. SR #7460 - If the picture of a COMP-3 or DISPLAY item specifies a signed quantity, then the VALUE clause will produce a value with C or D for the sign nibble or the appropriate overpunch.

7. SR #7495 - The $CONTROL NOMAP option was added to the compiler.

8. SR #7516 - If the compiler switched to virtual symbol table in the middle of a REDEFINES clause, then the redefining item's name could be changed to some third name. This could cause ERROR 21 or ERROR 63. (COBOL C only). This has been fixed.

9. SR #8131 - If a group preceded an item that needed a slack byte, then the slack byte was included in the previous group. This has been fixed.

10. SR #8271 - On a simple COMPUTE in an assignment of a subscripted item to another item, the result might be a power of 10 too large. This occurs if the subscripted item contains decimal places.

11. SR #8325 - READ INTO and RETURN INTO would always do the MOVE before checking for AT END or INVALID KEY. This has been fixed.

12. Level 88 can only contain 63 value pairs. This condition is now flagged with error 133.

13. SR #9199 - An ADD CORRESPONDING to table items that are COMP-3 will now work correctly.

14. SR #9294 - An ADD or SUBTRACT A,B to C where A and B are COMP and C is COMP-3 will no longer leave garbage on the stack causing stack overflows.

15. SR #10062 - The SEARCH ALL statement may now have a literal as the value in the WHEN condition.

16. SR #10204 - Level 88 items with a value clause subordinate to a valued group item are now allowed.

17. SR #10285 - A non-standard level number on a group item mades an empty group. Now the error message for empty group will also appear.

18. SR #10290 - A MOVE comp to an alphanumeric edited item caused a run-time bounds violation which has now been fixed.

19. SR #10356 - A group item of more than 120 characters that contains a value clause will no longer cause errors 201 and 207.

20. SR #10519 - Initialization of items or tables larger than 32k bytes is now performed correctly.

21. SR #10554 - MOVEs of non-numeric literals that contained only numerics to a numeric item sometimes produced error 164. This has been fixed.

22. SR #10639 - Programs with duplicate paragraph names, duplicate subordinate item names, many MOVE/ADD/SUBTRACT CORRESPONDING statements causes virtual symbol table problems that resulted in Error 403 - EOF ON COBSTAB, bounds violation, or undefined local paragraphs. The map would also be incorrect. (COBOL C only) Fixed.

23. SR #10836 - Error 200, USL FILE OVERFLOW, occurred even though the USL file directory was only half full. Fixed.

24. SR #11124 - PICTURE fields with P-factor did not correctly account for possible leading signs in the value clause causing Error 95 - NUMERIC LITERAL AND ITS PICTURE DISAGREE. This has been fixed.

25. SR #11226 - The RELEASE record FROM dataname statement could produce Error 157, SYNTAX REQUIRES DATANAME. Error 176 was not generated for incorrect READ INTO, RETURN INTO, RELEASE FROM, and WRITE FROM INVALID KEY. This has been fixed.

26. SR #11268 - A large program with many non-numeric compares produced Error 13, SPACE OVERFLOW when one word was left on the stack for each compare. Fixed.

27. SR #11518 - SR #6278 for duplicate paragraph names was fixed incorrectly causing Error 105 for legal duplicate paragraph names. This has been fixed.

28. The SEARCH ALL statement will now work correctly with expressions in the WHEN condition.

29. SR #10586 - Purging of the initialization segment from a USL file will now be listed for a LISTUSL command following the recompilation of the program into the same USL.

30. Error message 173 was issued instead of 169 on a READ with INVALID KEY. This has been fixed.

31. Sequence checking was stopped when EOF occurred on the masterfile or the masterfile was $NULL. (see manual example on page B-17) This has been fixed.

C. DOCUMENTATION CHANGES

1. Document flags appearing on the MAP, with an example. The following are flagged:

   a. COMP items on byte boundaries.

   b. COMP items over 9 digits.

   c. Unsigned COMP and COMP-3 items.

   d. COMP-3 items of even number of digits.

2. Document error 133 - Too many value pairs.

D. MISCELLANEOUS

1. The effects of slack byte changes.

   Besides correcting the case where slack bytes are inserted into the previous group, slack bytes will cause the starting location for groups to be moved so groups will not contain them. For example:

```
01
   02 PIC X.
      04
         06 COMP SYNC
```

   Before this change, only the 04 level was moved. Now all containing groups that begin at the same location will be moved so that the slack byte will be contained in as few groups as possible, i.e. the 01 level. This may affect some table definitions if something like the above example was used. Also if in the above example 02 was REDEFINEd then error 233 will be issued.

2. The effect of the DIVIDE fix.

   When the DIVIDE statement found that the operation would take more than 28 digits in the intermediate result, it would shift off digits in the dividend. Now it will correctly preserve the high order digits at the cost of low order digits. This may affect operations as follows:

```
  77 X PIC S9(15)V999 77 F PIC S9V999 VALUE 2.173.

COMPUTE X ROUNDED = F ** 10 / F / F / F / F / F
                                / F / F.
```

   The result used to be equal to F. Now it is 0. This operation should be broken up in steps or done in floating point.

SORT-MERGE/3000 HP32214B.02.03

NOON214B.HP32214.SUPPORT

A. ENHANCEMENTS

1. SORT-MERGE performance has been improved through the use of multirecord IO on the input and output files. Files passed to SORT/MERGE through the intrinsics may now be opened with either NOBUF or MULTIRECORD aoption. Sort will allocate the required buffer and perform deblocking for fixed or variable length files.

2. SORT-MERGE now supports ASCII or EBCDIC input data and ASCII, EBCDIC, or user defined collating sequences. User defined collating sequences can only be used if the data is ASCII.

3. SORT-MERGE now supports the use of MPE commands from the SORT-MERGE command processor.

4. SORT-MERGE will set the JCW to FATAL if an alternate output file (OUTPUT##) is created in a job stream.

5. The control-Y trap for SORT has been modified to print the number of records merged during the intermediate passes to yield an improved indication of the progress of SORT.

6. An EXIT command is now available so that you may leave SORT-MERGE/3000 without performing a sort or merge. You no longer need to BREAK and abort the program.

7. New MERGE intrinsics are available which are similiar to the current SORT intrinsics.

8. Multiple input files may now be specified in SORT.

B. CORRECTIVE SOFTWARE CHANGES

1. [SR# 6834] READ/WRITE errors could occur on the "SORTSCR" file if the file was overridden with a file equation and not built correctly. SORT now verifies that "SORTSCR" has been built correctly.

2. [SR# 7703] The VERIFY command would always report the record length to be 80 characters, even if the input file had a different record length. The VERIFY command will now report either "SAME AS THAT OF THE INPUT FILE" or the record length as modified by an INPUT command.

3. [SR# 7864] The JCW was always set to FATAL for syntax errors in commands. The JCW is now only set to fatal in stream jobs, not in interactive sessions where the commands may be corrected.

4. [SR# 8060] When END-OF-TAPE was encountered while writing the output file to unlabeled tape, SORT-MERGE would stop with a write error on the output file. SORT-MERGE will now rewind and unload the tape and prompt for the next tape.

5. [SR# 10191] After an incorrect KEY command containing multiple keys was entered, the keys array was not correctly restored. The error recovery for the KEY statement now ignores the entire KEY command.

C. DOCUMENTATION CHANGES:

A new SORT-MERGE manual is available (32214-90001).

A. ENHANCEMENTS

1. DBML changes:

a. The uncallable procedure DBLOCKINFO has been modified so that the IMAGE SIR is held for as short a time as possible. This reduces the impact on users whenever a SHOW ALL or SHOW LOCKS or SHOW USERS is performed by DBUTIL.

b. The uncallable procedure DBABORT has been modified so that the creation of I-files due to a critical system failure will occur only if DUMPING has been ENABLEd for the affected data base. (See DBUTIL enhancement 2.f below).

c. A new uncallable procedure DBMAINT has been added. This procedure is driven by DBUTIL and is used to CREATE or ERASE data bases. This function was formerly performed by DBOPEN. DBMAINT differs from DBOPEN in not using the IMAGE SIR and in using multi-record FWRITEs to initialize data sets. Not using the IMAGE SIR allows other IMAGE users to be processed concurrently. This initialization method avoids usurping the disc driver and minimizes the impact on all other processes. The creation/erasure is also significantly faster for very large data sets. Achieving these advantages was made possible by forcing the record size of each data set to be a multiple of 128 words. The disc requirements remain unchanged. DBMAINT uses a temporary extra data segment. This data segment is 'tagged' with "IMAGE4" in its first six (6) bytes.

2. DBUTIL has the following new features:

a. A data base RELEASE command

RELEASE data base name

by means of which the creator can RELEASE the root file and data set files of a data base.

This applies the same function as the :RELEASE command of MPE to all the files of an IMAGE/3000 data base.

b. A data base SECURE command

SECURE data base name

by means of which the creator can SECURE the root file and data set files of a data base.

This applies the same function as the :SECURE command of MPE to all the files of an IMAGE/3000 data base.

c. A new entry point called OLDWAY has been added. If the user issues a CREATE or ERASE after having entered DBUTIL via the OLDWAY entry point, the CREATE or ERASE will be performed via DBOPEN as under previous versions This has been provided in case any problems are encountered with the new way which is done via DBMAINT.

   WARNING This entry point will eventually be deleted.

d. The SHOW command has been augmented with a PASSWORDS option:

   SHOW data base name[/maint word] PASSWORDS

   which enables the creator, or a user providing the correct maint word, to display the 63 access classes and their passwords.

   This was implemented in an earlier release but undocumented in the earlier NOON file.

e. The SET command has been augmented with a PASSWORD option:

   SET data base name[/maint word] PASSWORD n=password

   which enables the creator, or a user providing the correct maint word, to assign a new password to the access class n.

f. The ENABLE and DISABLE commands have been extended to support a DUMPING flag. When DUMPING is DISABLED, no I-file is created by DBABORT (this is the default condition of all data bases). When DUMPING is ENABLED, an I-file is created by DBABORT. The commands are:

   ENA[BLE] data base[/maint word] FOR DUMPING

   and

   DIS[ABLE] data base[/maint word] FOR DUMPING

   The SHOW command has also been extended to display the state of this DUMPING flag.

   In general, the I-file created by DBABORT is a debugging tool for testing changes to IMAGE within a controlled environment. For this reason, DUMPING should normally be DISABLED.

3. DBLOAD has been modified to ensure having an adequate number of buffers to provide good performance. This was done in the form of an over-ride to the buffer specs in case the buffspecs established by the user for application efficiency were less than needed for good performance during loading.

4. DBDRIVER has been modified to display the version/update/fix level of DBMAINT. DBMAINT is the uncallable procedure which is used by DBUTIL to CREATE and/or ERASE data bases. The PRIV user of DBDRIVER can use DBDRIVER to CREATE/ERASE data bases. With mode set to 1 (CREATE) or 2 (ERASE) and with base and qualifier set, respond to the "COMMAND: " prompt with Z.

5. File equates employed with DBUNLOAD, DBLOAD, DBSTORE and DBRESTOR may now include a filename part.

   Example: :FILE DBLOAD=GEORGE;DEV=TAPE

   When DBLOAD is RUN, the string "GEORGE" will appear in the console prompt instead of "DBLOAD".

   This was implemented in an earlier release but not documented in the earlier NOON file.

6. DBUNLOAD has been modified to continue in spite of broken chains (during a chained unload). It unloads (saves) all entries in the chain up to (but not including) the break, and all entries from the chain end. In some cases, all entries in the chain may be saved. In all cases, the order of the entries is preserved.

   The occurrence of each broken chain results in a multi-line message which:

   - identifies the Entry number where the break was detected
   - identifies the number of the Entry (last one saved from the front of the chain) if any
   - identifies the Entry number and Data Set number of the chain head
   - displays the value of the key of the broken chain
   - displays the chain length and the number of entries salvaged

   The end-of-data set summary reports the number of lost entries, if any.

A 'dummy' DBUNLOAD can be made by using the FILE command

```
:FILE DBUNLOAD=$NULL
```

before running DBUNLOAD.

B. CORRECTIVE SOFTWARE CHANGES

1. SR #9135 DBDRIVER permits non-privileged users to PURGE an IMAGE I-file created by DBABORT.

2. SR #9136 DBRESTOR permits starting with other than the first volume of a multivolume STORE or SYSDUMP tape.

3. SR #9324 DBGET returns CW=18 when an empty record is encountered as a result of a broken chain.

4. SR #9914 DBUTIL handles maint words correctly when run from the ERASE or PURGE entry points.

5. SR #9922 DBSTORE correctly reports the file system error number when failure is due to a tape error.

C. MISCELLANEOUS

IMAGE/3000 does NOT support labelled tapes. In particular, DBSTORE does NOT write labelled tapes and DBRESTOR does NOT read labelled tapes.

This version of IMAGE replaces release 32215B.02.03.

Systems running versions of MPE II should continue to use the lastest version of IMAGE A.

A. CORRECTIVE SOFTWARE CHANGES

1. SR# 7829 A QUERY report containing headers and the output control statement, LINES = 0, would abort with a stack overflow. With no headers and LINES = 0 QUERY would page after every line that it printed. This condition has been fixed.

2. SR# 10840 QUERY has been fixed so that it will now accept signed positive or negative numeric literals, such as "-4" or "+166", in register statements instead of giving the message:
   INVALID NUMERIC DIGIT

3. SR# 9613 If a report with no SORT statements was executed following a report with multiple sort statements, the second sort failed with loss of data. This error has been fixed.

4. SR# 8431 Dividing by zero previously resulted in the quotient being set to the numerator, rather than zero. QUERY now returns a zero quotient for a divide by zero.

5. SR# 9848 Previously, QUERY's REPORT command did not print headers if none of the following was included:
   a. detail statement
   b. group statement
   c. total statement (excluding TF)
   This condition has been fixed.

6. SR# 7556 QUERY has been changed so that it no longer aborts in job mode if the lockword supplied by the user for an XEQ file is invalid.

7. SR# 10202 Previously, when doing a retrieval on a master data set of the form:
   FIND SEARCH-ITEM = VALUE1, VALUE2, ...,VALUEN
   QUERY responded:
   "NO ENTRY"
   for every value not having a corresponding entry. If any entries qualified, it then said:
   "N ENTRIES QUALIFIED"
   Due to known problem SR #7425 (fixed in version A.04.01 of QUERY), where QUERY aborted in job mode for non-serious errors, QUERY aborted on the "NO ENTRY" condition, even if some entries qualified. This problem has been fixed.

8. SR# 10339 The QUERY manual states that all insertion characters in an edit mask will be printed. However, if the edit mask had insertion characters to the right of the position in the edit mask in which the last character of the literal data item would be printed, they were replaced by asterisks. This problem has been fixed.

B. DOCUMENTATION CHANGES

1. Page 4-16, the syntax of the header statement should be as follows:

```
H header number, print element, print position
[,SPACE A [number] ] [,SPACE B [number] ]
[,E {number} ]
    {  Z   }
```

   allowing the use of edit masks with header statements.

2. SR# 11100 The following error message:
   DATA BASE NOT ENABLED (SUBSYSTEM ERROR)
   is not documented in the QUERY manual. This error message will only occur as a result of a bug which has been fixed in version A.04.01 (SR# 7424). The problem is that no updates can be made after the assignment of LOCKOPTION=OFF. The above error message results if an update is attempted while LOCKOPTION is in the OFF state.

3. SR# 9065 Page 4-45b, add the following item:
   5. When OUTPUT=TERM then a page eject is defined as skipping 6 lines.

4. Page 2-10, in the description of the example that is given on that page, "INVENTORY data base" should be changed to "INVENTORY data set".

5. Page 3-23, in the description of the example that is given on that page, the word "equal" should be changed to "greater than".

XA2100/3000 HP32223A.01.04

NOON223A.HP32223.SUPPORT

CORRECTIVE SOFTWARE CHANGES

SR #6470

When assembling source files with XA2100, records whose length exceeded that of the listing device would not be listed, and a file error message would be generated. It is now possible to use listing devices with widths as low as 30 characters. For ease of reading the listing, however, it is recommended that the list width exceed 50 characters. Long titles will be truncated, but all other records will be listed in their entirety.

COBOL LIBRARY HP32232A.00.01

NOON232A.HP32232.SUPPORT

A. ENHANCEMENT

The COBOL LIBRARY now includes the COBOL II runtime library procedures.

B. CORRECTIVE SOFTWARE CHANGES

1. SR #7213 - Zero may be raised to positive powers.

2. SR #9050 - If the stack is greater than 32K bytes, then the SELECT ... ASSIGN to "$STDLIST" statement would not default to CCTL if the $STDLIST device was a terminal. (COBOL B and C only) This has been fixed.

3. C'QUIT was added to allow users to enter DEBUG when the :SETDUMP command is issued before running a COBOL program. (COBOL B and C only)

4. Error 711 - "ILLEGAL SOURCE DIGIT IN CONVERSION" will be printed when zoned overpunches occur in positions other than the right-most digit of a display item and the overpunches will be ignored. Previously, the zone

overpunches were ignored, but no error message was printed. Only one error message per data item will be generated instead of one message per digit.

5. For COBOL II, a WRITE after a successful READ statement on a sequential file that is opened I/O will update the record that was read, instead of the next record in the file. (COBOL II only)

6. Signed DISPLAY would have the right-most digit that was negative 0 become an unsigned 0 when using an EXAMINE statement. This has been fixed.

7. Unaligned or COMP items with more than 4 digits will now work for the AFTER ADVANCING options of a WRITE statement. (COBOL B and C only)

8. MOVE TIME-OF-DAY to a signed display item would always be unsigned. (COBOL B and C only) This has been fixed.

9. The ALPHABETIC class condition will now treat upper and lower cases equivalently.

10. The library will now handle unaligned COMP items in all cases.

11. SR #10290 - A MOVE comp to an alphanumeric edited item caused a run-time bounds violation which has now been fixed.

CORRECTIVE SOFTWARE CHANGES

The following problems have been corrected:

1. SR #11322 - $CONTROL ERRORS= would ignore the first digit following the "=".

2. SR #11323 - A COMPUTE statement with the ROUNDED option which contains a division operator would have the result truncated instead of rounded.

3. SR #11493 - The SORT statement would ignore any file equation for the SORT SD file in determining how many records to be sorted, and only would use the value from the ASSIGN clause.

4. SR #11436 - The EXAMINE statement would not work correctly for numeric edited items because the length of the data item was calculated incorrectly.

5. SR #11537 - The SORT statement with an input procedure would cause any procedure currently being PERFORMed to not return. Also any collating sequence for the SORT would be offset by 2 bytes.

6. Identifier-1, word-1 or the reserved words "OF" "IN", and "BY" in the COPY REPLACING statement would not be recognized correctly if they were in lower case.

7. The $CONTROL VERBS would not print any verbs if they occurred between a section and the first paragraph name.

8. For certain "disastrous" compile-time file errors, no error listing and no compilation statistics would be printed.

N00N230A

Release issue of HP 32230A Series II/III diagnostics.

Magnetic tapes associated with HP32230A:

Source	32230-1X001
CPU Coldload	30000-1X016
NON-CPU C/L	30000-1X017

Manuals associated with HP32230A:

32230-60001
32230-60002

*** CPU ***	30000-1X016 1906	
SECTION 1	PD420A	01.00
SECTION 2	PD420A1	01.00
SECTION 3	PD420A2	01.01
SECTION 4	PD420A3	01.03
SECTION 5	PD420A4	01.00
SECTION 6	PD420A5	01.00
SECTION 7	PD420A6	01.00
SECTION 8	PD420A7	01.00
SECTION 9	PD420A8	01.00
SECTION 10	PD420A9	01.00
SECTION 11	PD420A10	01.00
SECTION 12	PD420A11	01.00
SECTION 13	PD420A12	01.01
SECTION 14	PD420A13	01.00

*** STAND-ALONE ***	30000-1X017 1949		% FILE NO.
SLEUTH	PD411A	01.04	(01)
SDUPII	PD417A	01.03	
CART DISC-7905A	PD419A	01.04	(02)
MEMORY PATTERN	PD421A	01.00	(03)
MULTIPLEXOR CHAN	PD422A	01.02	(04)
DISC FILE-2888A	PD423A	01.00	(05)
CART DISC-7900A	PD424A	01.00	(06)
SYSTEM CLOCK	PD425A	01.00	(07)
SYS CLK/FLI	PD426A	00.00	(10)
TERMINAL DATA	PD427A	01.01	(11)
FIXED HEAD DISC	PD428A	01.00	(12)
SELECTOR CHAN	PD429A	01.01	(13)
FAULT CORRECTING MEM.	PD430A	01.01	(14)

*** STAND-ALONE *** 30000-1X017 1949 % FILE NO.

MEMORY DIAGNOSTIC	PD430B	00.01 **	(15) FIXED
EXTENDED INSTRUC SET	PD431A	01.00	(16)
HSI DIAG.	PD432A	01.00	(17)
MAGNETIC TAPE	PD433A	01.04	(20)
SSLC INTERFACE	PD434A	01.03	(21)
ASLC INTERFACE	PD434B	01.04	(22)
UI DIAG	PD435A	01.01	(23)
SPECIAL HSUI DIAG	PD436A	00.00	-----
TERMINAL CONTROL	PD438A	01.00	(24)
CALCOMP PLOTTER	PD439A	01.01	(25)
COBOLII(PART 1)	PD441A	00.00 **	(26) NEW RELEASE
COBOLII(PART 2)	PD442A	00.00 **	(27) NEW RELEASE

*** ONLINE ***

CARD READER	PD465A	01.00
LINE PRINTER	PD466A	01.02
2617j line printer	pd466j	01.00
2640 TERMINAL	PD469A	01.00
TERM-2635A	PD474A	00.00
TERM-2762A/B	PD475A	01.00
term-2645k	pd476a	00.00
DISPLAY TERMINAL 2644	PD477A	01.00
TERM-2615A	PD478A	01.00
CARD-READ/PUNCH	PD479A	01.00
OPTICAL MARK READER	PD480A	00.00

UTILITY FILES

SLEUTH BATCH FILES

```
**************************************************
*                                                *
*  THESE FILES MAY ONLY BE USED IN CONJUNCTION   *
*  WITH THE SLEUTH PROGRAM. REFER TO THE SLEUTH  *
*  MANUAL FOR INFORMATION ON HOW THEY MAY BE     *
*  LOADED.                                       *
*                                                *
**************************************************
```

FILE NAME	FUNCTION
SLEUTH01	
SLEUTH02	
SLEUTH03	
SLEUTH04	

FILE NAME	FUNCTION
SLEUTH05	
SLEUTH06	
SLEUTH07	DISC VERIFIER-7905,7906,7920,&7925
SLEUTH08	
SLEUTH11	LONG CARD READER DIAG-SECTION1
SLEUTH12	LONG CARD READER DIAG-SECTION2
SLEUTH13	LONG CARD READER DIAG-SECTION3
SLEUTH14	LONG CARD READER DIAG-SECTION4

STAND-ALONE DIAGNOSTIC TAPE CREATORS

```
**********************************************
*                                            *
* THESE FILES ARE STREAMABLE JOB FILES WHICH  *
* WILL CREATE CONFIGURED CPU DIAGNOSTIC TAPES *
* AND NON-CPU DIAGNOSTIC TAPE.                *
*                                            *
**********************************************
```

FILE NAME	FUNCTION
CPU064	CPU TAPE CONFIGURED FOR 64K OF MEMORY
CPU096	CPU TAPE CONFIGURED FOR 96K OF MEMORY
CPU128S2	CPU TAPE CONFIGURED FOR 128K OF MEMORY
CPU128S3	CPU TAPE CONFIGURED FOR 128K OF MEMORY
CPU160	CPU TAPE CONFIGURED FOR 160K OF MEMORY
CPU192	CPU TAPE CONFIGURED FOR 192K OF MEMORY
CPU224	CPU TAPE CONFIGURED FOR 224K OF MEMORY
CPU256S2	CPU TAPE CONFIGURED FOR 256K OF MEMORY
CPU256S3	CPU TAPE CONFIGURED FOR 256K OF MEMORY
CPU384S3	CPU TAPE CONFIGURED FOR 384K OF MEMORY
CPU512S3	CPU TAPE CONFIGURED FOR 512K OF MEMORY
CPU640S3	CPU TAPE CONFIGURED FOR 640K OF MEMORY
CPU768S3	CPU TAPE CONFIGURED FOR 768K OF MEMORY
CPU896S3	CPU TAPE CONFIGURED FOR 896K OF MEMORY
CPU1KS3	CPU TAPE CONFIGURED FOR 1024K OF MEMORY
DIAGIOTP	NONCPU TAPE (%27 FILES SEE ABOVE FOR NEW FILE REFERNCE TABLE)

SUPPLEMENTAL FILES FOR DIAGNOSTICS

```
**********************************************************
*                                                        *
*  THESE FILES ARE REQUIRED BY THE INDICATED DIAGNOSTIC  *
*  TO OPERATE PROPERLY.                                  *
*                                                        *
**********************************************************
```

FILENAME	DIAG NO.	FUNCTION
VFCTEST	D466A	DATA FILE FOR 2608 LP
STDVFC	D466A	DATA FILE FOR 2608 LP

** IMPLIES FIXED THIS TIME

****

FIX LEVEL .01 MEMORY DIAGNOSTIC

D430B.00.01

THIS FIX CORRECTS A PROBLEM ENCOUNTERED WHEN CHECKING A BUFFER OF DATA DURING THE DIAGNOSTIC.

****

FIX LEVEL .00 COBOLII FIRMWARE DIAGNOSTIC PART 1

D441A.00.00

THIS IS A NEW RELEASE. THE DIAGNOSTIC CHECKS THE FIRMWARE CHIPS WHICH ARE ADDED TO THE EIS BOARD. THIS IS PART ONE OF TWO.

****

FIX LEVEL .00 COBOLII FIRMWARE DIAGNOSTIC PART 2

D442A.00.00

THIS IS A NEW RELEASE. THE DIAGNOSTIC CHECKS THE FIRMWARE CHIPS WHICH ARE ADDED TO THE EIS BOARD. THIS IS PART TWO OF TWO.

****

MPE III Series 30-33 Software Update

MULTIPROGRAMMING EXECUTIVE OPERATING SYSTEM SERIES 33
****

CONTENTS OF INSTALLATION TAPE DATE CODE '2011'

PRODUCTS WITH ASTERISKS ARE THE PRODUCT(S) UPDATED/CHANGED BY THIS M.I.T. AND ALSO REFERENCE PERTINENT NOTE FILES CONTAINING INFORMATION ABOUT THE MODIFICATIONS. THESE FILES MAY BE LISTED USING EDITOR OR FCOPY.

****

PRODUCT NAME	PRODUCT NUMBER	LEVEL	DATE CODE
*MPE	32033B	01.01	2011
*SEGMENTER	32050A	01.02	2011
*SPL	32100A	08.00	2011
*BASIC	32101B	00.12	2011
*FORTRAN	32102B	01.03	2011
*BASIC COMPILER	32103B	00.12	2011
*RPG	32104A	04.05	2011
BUILDINT	32150A	03.01	1623
*DS/3000	32190A	02.05	2011
*EDITOR	32201A	07.07	2011
SCIENTIFIC LIBRARY	32205B	00.04	1906
*DEL/3000	32206A	01.10	2011
*KSAM/3000	32208A	03.01	2011
*V/3000	32209A	01.01	2011
*COMPILER LIBRARY	32211D	00.10	2011
*FCOPY	32212A	03.11	2011
*COBOL	32213C	02.05	2011
*SORT/MERGE	32214B	02.03	2011
*IMAGE	32215B	02.05	2011
*QUERY	32216A	04.03	2011
*DIAGNOSTICS	32231A	-- --	-- --
*COBOL LIBRARY	32232A	00.01	2011
*COBOLII	32233A	00.01	2011
RJE 2780/3780	30130E	00.03	DATACOMM

DIAGNOSTIC INFORMATION IS CONTAINED IN THE FILE NOON230A.

* NOTE FILES(NOONYYYZ) CONTAIN THE CHANGE INFORMATION

WHERE YYY =LAST THREE DIGITS OF THE PRODUCT NUMBER.
(E.G. MPE IS HP32002. THEREFORE YYY=002.)
Z =CURRENTLY RELEASED VERSION DIGIT OF PRODUCT.

I. MPE 32033B.01.01

A. MODULES MODIFIED B.01.01

MODULE		CHANGE HISTORY		
NAME	NO	A.00.XX	B.01.XX	
		0	0	1 2 3 4 5 6 7 8 9
ININ	10	X	X	X
IOTAPE0	18	X	X	X
IOLPRT0	19	X		X
IOLPRT1	21	X		X
IOTERM0	22	X	X	X
IOFLOP0	23	X		X
IOMDISC1	27	X		X
PFAIL	30	X		
* SDFCHECK	33	X		
* SDFLOAD	33	X		
SDFGEN	34	X		
HARDRES	55	-	X	X
NRIO	62	X	X	X
MEASIO	88	-	X	X

* Both program files are generated as part of module 33

NOTE: These module are all MPE modules which differ in code and source between the Series II/III and the Series 33. For changes to modules which are common to both the Series II/III and the Series 33, see the note file for the Series II/III.

SYSTEM	LAST CHANGE NUMBER
A.00.00	0485
A.01.00	0709
B.01.01	1248

B. ENHANCEMENTS

FIX NUMBER	DESCRIPTION
1004.	HARDRES (55) Allows aborts while using TEPE/3000.
1040.	ININ (10) DEBUG now handles COBOL74 two word instructions.
1045.	HARDRES (55) Added support to TELENET (Termtype 13) to the ADCC driver.
1046.	IOTERM0 (22) Added support to TELENET (Termtype 13) to the ADCC driver.
1183.	ININ (10) Changed the Power on procedure so that it will handle the system clock interface correctly.
1184.	HARDRES (55) Changed procedures so that it will facilitate changing clock interrupt interval.
1185.	MEASIO (88) Added procedures so that it will set and reset the system clock at different interrupt intervals.
1201.	IOLPRT0 (19) This driver now supports function code 15 (status request).

FIX NUMBER	DESCRIPTION

1201. IOMDISC1 (27)
This driver now supports function code 15 (status request).

1218. HARDRES (55)
A new terminal type of 18 has been created. This terminal type will not send a DC1 when a read is issued, does not use the ENQ/ACK protocol, and does not send any sync characters after line feeds or carriage returns. It will operate at the following speeds: 10,15,30,60,120,240,480,& 960 char/sec.

C. CORRECTIVE SOFTWARE CHANGES

848. IOTERM0 (22)
This fixes the problem when phone line is disconnected, but session is not aborted.

848. HARDRES (55)
(See IOTERM0 description)

860. HARDRES (55)
Attachio uses "ASMB(XCH,SUB)" to calculate distance from Q to DL as Q-Rel index. This can cause integer overflow in top 1/2 of a bank. Change to "ASMB(LSUB NEG)" to achieve same results without overflow problem.

926. HARDRES (55)
This change will make consecutive calls to the timer consistant with respect to each other. Timer will not appear to move forward or backward in relation to successive timer calls.

1006. HARDRES (55)
Fixes RESTART write problem with 264X when ENQ is sent after an Xoff.

1134. HARDRES (55)
Fixed a console lock out problem in which the system disc was not ready and "LDEV #1 NOT READY" was printed.

FIX NUMBER	DESCRIPTION
1202.	NRIO (62) FCONTROL 3 will now return a single device status word.
1226.	IOLPRTO (19) Fixed to handle prespace mode on offline condition so it will not lose a line.
1248.	NRIO (62) Reset SYSDB for invalid devices in device status, else system fail.

N00N231A

Release issue of HP 32231A Series 30/33 diagnostics
and utilities.

Cartridge tapes associated with HP32231A:

Maintenance Interface and Cold Load Self Test	30070-10401
Maintenance Display Software	30070-10402
Remote Maintenance/Console Facility	30070-10403

Flexible disc associated with HP32231A:

Diagnostic and Utility System	30070-13401

Manual associated with HP 32231A: 30070-60068

Utilities on MIT: DUSCOPY Version 0.02

*** Maintenance Interface and ***
*** Cold Load Self Test ***
** 30070-10401 Rev 2001 **

Maintenance Interface Diagnostic Version 0.03 **
Cold Load Self Test Version 0.03 **

TO GENERATE AN MI DIAGNOSTIC AND CLST TAPE
ON THE RIGHT CARTRIDGE TAPE,
RUN FCOPY.PUB.SYS AND ENTER THE FOLLOWING COMMANDS:

```
>FROM=MIDHEAD;TO=$CTUR
>FROM=MIDLBINS;TO=$CTUR;SKIPEOF=,2
>FROM=CLSTHEAD;TO=$CTUR;SKIPEOF=,3
>FROM=CASET4;TO=$CTUR;SKIPEOF=,4
>EXIT
```

*** Maintenance Display Software Cartridge Tape ***
** 30070-10402 Rev 1949 **

Maintenance Display Software Version 0.02 **

TO GENERATE A MAINTENANCE DISPLAY TAPE
ON THE RIGHT CARTRIDGE TAPE,
RUN FCOPY.PUB.SYS AND ENTER THE FOLLOWING COMMANDS:

```
>FROM=MPHEAD;TO=$CTUR
>FROM=MPLINKBS;TO=$CTUR;SKIPEOF=,2
>EXIT
```

*** Remote Maintenance/Console Facility Cartridge Tape ***
** 30070-10403 Rev 1949 **

Remote Maintenance/Console Facility Version 0.03 **

TO GENERATE A REMOTE TAPE ON THE RIGHT CARTRIDGE TAPE,
RUN FCOPY.PUB.SYS
AND ENTER THE FOLLOWING COMMANDS:
>FROM=REMHEAD;TO=$CTUR
>FROM=REMLINKB;TO=$CTUR;SKIPEOF=,2
>EXIT

*** Diagnostic and Utility System Flexible Disc ***
** 30070-13401 Rev 2003 **

Diagnostic Utility System		Version 0.36 **
ININ	PD570A	
SADS	PD590A	
AID	PD550A	Version 0.32 **
ADCCDIAG	PD509A	Version 0.10
GICDIAG	PD508A	Version 0.19
MEMDIAG	PD507A	Version 0.05
SADUTIL	PD575A	Version 0.06 **
IOMAP	PD560A	Version 0.05 **
D7902	PD513A	Version 0.04
D13037	PD511A	Version 0.05 **
D7970S13	PD514A	Version 0.09 **
D7970S45	PD515A	Version 0.05 **
D7970S68	PD516A	Version 0.05 **
VERIFIER	PD512A	Version 0.06 **
SLEUTHSM	PD555A	Version 0.06 **
D7910K	PD517A	Version 0.01 **
RATES	PD519A	Version 0.02 **
PRMDIAG	PD518A	Version 0.00 **

This note describes how to create the DUS flexible disc. A binary image of the DUS flexible disc is in the file DUS.HP32231.SUPPORT. This image may be placed on a previously formatted flexible disc (must be serializedlized see note below) by running the program DUSCOPY. HP32231.SUPPORT. This program writes the flexible disc and then reads back the data to ensure that the data is correct.

NOTE: Serializing a Flexible Disc.

Once a flexible disc has been formatted it can be serialized in the following manner:

a) On the system console, down the ldev for the flexible disc.
b) Log on and enter VINIT. Install flexible disc.
c) When prompted (>) enter - serial (ldev) i.e serial 3.
d) The program will return another prompt (>) when complete. Enter exit to end the program.
e) On the system console - up the flexible disc (up ldev).

NOTE: A diskette can also be formatted with VINIT by entering FORMAT (ldev).

****
****

This note file contains all change history since the original note file (1912). Changes indicated by "**" above are since 1918.

****

Fix Level .02 DUSCOPY

Change made to allow DUSCOPY to be run from any account.

****

Fix Level .03 Maintenance Interface Diagnostic

Version 0.03

The delay for the front panel one shots and switch debouncers in the DUMP and LOAD tests (17 and 18) is increased.

All HP-IB interface tests have been removed.

Linker modified to allow proper loading and interaction of all programs (MI, CLST and MD) with REMOTE.

****

Fix Level .12 Cold Load Self Test

Version 0.05

This fix allows the user to run at 80ns which occurs when running with external clock (an option on the maintenance display).

Version 0.06

Those GIC tests used on HP300 system self test but previously missing on Series 33 have been added. Those tests which used the MI - HP-IB cable have been deleted. (The cable is no longer used and can be designed out of the system.)

Error condition set when reading status is now checked correctly. This caused the test to run without error if CPU power was removed after the test started.

Version 0.10

Step 20 changed to allow two seconds instead of one second before timeout occurs.

Linker modified to allow proper loading and interaction of all programs (MI, CLST and MD) with REMOTE.

Version 0.11

Control store CRC testing time is changed from 3.5 seconds for steps 30-57 to 5 seconds.

Version 0.12

Allows program to continue after errors in CPU tests.

****

Fix Level .02 Maintenance Display

Version 0.01

This fix allows the user to set the RAR to all addresses including those on the firmware board.

Version 0.02

Linker modified to allow proper loading and interaction of all programs (MI, CLST and MD) with REMOTE.

****

Fix Level .01 REMOTE

Version 0.01

Corrected the escape "?" sequence that enters the remote comment mode. User can now enter this mode by pressing ESC "?" once. The "?" will only appear on the screen when entered from the remote console.

Eliminated the check for parity. Allows remote console to be in none or even parity positions.

Added line feed after detection of the carriage return character.

Changed from low priority character detection to high priority character detection. Characters are offered to other programs before REMOTE during low priority detection.

Linker modified to allow proper loading and interaction of all programs (MI, CLST and MD) with REMOTE.

****

****

Fix Level .36 DUS - ININ
- SADS

Version 0.35 (Only SADS changed)

DUS now auto-recognizes the HP300 flexible disc (DUS) and prints a warning message. This allows transfer of AID programs between families. 110 and 150 baud operation was corrected. Control Y will always cause a break when the driver is writing at 9600 baud.

Version 0.36 (Only ININ changed)

The timer interrupt routine is rewritten and moved from STT'30 to STT'12 of segment 1. This corrects STARTCLOCK problem reported in SMR 7779.

****

Fix Level .32 AID

PD550A Version 0.32

GETBOUNDS routine is changed to not allow the first number to be less than one. This fixes the problem reported in SMR 8125 where D 56100/5105 was input.

DB is changed to repeat string throughout string array when the pattern does not fit an exact number of times. This fixes the problem reported in SMR 7631.

****

Fix Level .06 Sadutil

PD575A Version 0.05

This version fixed the CPVA-magnetic tape problem seen when the magnetic tape was device zero.

PD575A Version 0.06

COND(ense) operation is changed to clear a bit in the free space table when the table becomes smaller. When the table becomes full, the operator is notified of this condition and the bit is set to reflect that the message was sent. When any kind of condense operation is run which results in a smaller free space table, the bit is cleared.

****

****

Fix Level .05 IOMAP

PD560A Version 0.05

Changed to add INP.

****

Fix Level .05 D13037

PD511A Version 0.03

Step 152 was added to test the 12745 buffer. Step 148 was changed to not test clock offset. Clock offset was previously deleted in the 13037 controller.

PD511A Version 0.04

Step 7 corrected to send each possible byte.

Program ends if no units are ready in step 10.

Step 119 changed to request the user to cycle power with HP-IB cable disconnected.

PD511A Version 0.05

Program changed to properly handle errors detected by AID and reflected in the NORESPONS reserved variable.

****

Fix Level .09 D7970S13

PD514A Version 0.06

Step 119 was changed to expect only one interrupt on REWIND from BOT. Step 226 was changed to expect only one interrupt after read record. Step 310 was changed to expect DSJ=1 when multi-error was simulated. Step 229 was changed to provide more time between read bursts.

PD514A Version 0.08

CLEARDEV sequence changed to match new firmware and called before each section.

Status is now read correctly when data transfer is performed with addressoff.

Duplicate status read corrected for write,read and read back.

Rewind rewritten for new firmware.

Rewindoff rewritten for new firmware.

Step 119 changed for new firmware.

Changed to work correctly with multi-units.

PD514A Version 0.09

Step 9 is changed to allow tapes without write rings. A "NO WRITE RING" message is added.

Error messages can now be output to the line printer. A new message is output in step 9.

Step 220 is changed to correctly use EOT instead of EOF.

****

Fix Level .05 D7970S45

PD515A Version 0.03

Step 431 was changed to repeat WR,BSR,RR sequence 400 times instead of 4000 times to reduce run time.

PD515A Version 0.04

CLEARDEV sequence changed to match new firmware and called before each section.
Status is now read correctly when data transfer is performed with addressoff.

Duplicate status read corrected for write,read and read back.

Portions which transfer data are shortened to decrease run time (steps 431,512,515,525,529,547 and 548).

PD515A Version 0.05

Step 9 is changed to allow tapes without write rings. A "NO WRITE RING" message is added.

Error messages can now be output to the line printer. A new message is output in step 9.

****

****

Fix Level .05 D7970S68

PD516A Version 0.03

Step 628 was changed to expect only one interrupt after REWIND from BOT. Step 624 messages were changed to check for RESET light. Section 8 measuring of feat was corrected.

PD516A Version 0.04

WRITEFM corrected.

CLEARDEV sequence changed to match new firmware and called before each section.

Status is now read correctly when data transfer is performed with addressoff.

Duplicate status read corrected for write,read and read back.

Rewind rewritten for new firmware.

Rewindoff rewritten for new firmware.

Step 617 changed to select correct unit at the end.

Step 625 changed to ask user if master is being used.

Step 802,804,807 and 810 changed to improve accuracy.

PD516A Version 0.05

A tempory patch was left in step 4. This patch has been deleted.

Step 802 is changed to issue CLEARDEV between one foot writes.

The write record function is changed to be the same as the other two magnetic tape diagnostics.

****

Fix Level .06 VERIFIER

PD512A Version 0.05

Table added to automatically show the user all flagged tracks.

Line printer capability added.

Ability to flag a track defective added.

PD512A Version 0.06

Added ability to flag a track defective after user has formatted the disc pack. Testing for flagged tracks is omitted in this case.

****

Fix Level .06 SLEUTHSM

PD555A Version 0.05

Program is revised to provide room for the addition of 7910 disc.

FMT function is modified to allow user to create and delete invisable tracks for a 7902 flexible disc.

Error messages are changed from PRINT to PRINTEX format.

WD for printers changed to improve page control. Printer is left at the top of form at the end of this function.

CHB function changed to not write into the word following the buffer when the decrement buffer option is selected.

PD555A Version 0.06

Corrected the select unit command for a Mag Tape in the DEV function. This function will only check for on line status now.

Revised the following functions to use common code:
RSA, CL, POLL AND RSYN

Corrected an error modifying the mask word for status word 2 on disc read errors.

Added target address info on all disc seek errors.

All disc errors will have the cyl,hd & sect defined as being the discs internal address.

Added a check for BOT and on-line status for REW function. When a REW function was issued to a mag tape already on load point (BOT), the previous Sleuthsm would indicate an error.

Timeout for rewind function is increased to approximately 2.25 minutes.

****

****

New Release D7910K

PD517A Version 0.01

This program included in case 7910 disc is ever offered.

****

New Release RATES

PD519A Version 0.02

This program included in case 7910 disc is ever offered.

****

New Release PRMDIAG

PD518A Version 0.00

This new diagnostic supports new hardware.

****

DOCUMENTATION

The catalog of customer publications at the end of this section lists the currently available customer manuals for HP 3000 Computer Systems products. This list supersedes the catalogs in previous issues of the COMMUNICATOR.

Purchasing

Customers may purchase copies of new manuals, new editions and updates by either Direct Phone Order or by placing orders through their local HP Sales and Service Office.

The Direct Phone Order numbers are (800) 538-8787 (toll free) and, in California, (408) 738-4133 (collect). Calls should be made between 9:00 a.m. and 5:00 p.m. in the caller's time zone. Most orders will be shipped within 24 hours.

The addresses and telephone numbers of local HP Sales and Service Offices are listed in the back of all customer manuals.

Prices of HP documentation are subject to change without notice.

To obtain a manual update, the customer must purchase the manual to which it pertains. The latest edition of the manual, along with the update, will then be sent to the customer.

Terms

A few words about documentation terms and procedures.

NEW — The first printing of the first edition. When first printed, a manual is assigned a part number that is retained for the life of the manual.

UPDATE — A supplement to an existing manual which contains new or changed information. Manual updates, which are issued between editions, contain additional or replacement pages to be merged into the manual by the customer.

Updates are generally issued at the same time Installation Tapes (ITs) are issued. However, THERE IS NO DIRECT CORRELATION BETWEEN SOFTWARE FIXES AND MANUAL UPDATES. Software enhancements that require documentation changes will be accompanied by manual updates, but software fixes and manual corrections may be made independently.

Updates are retroactively inclusive; that is, whenever successive updates are issued, the later update will contain the previous one(s). This means that you need obtain only the latest update to have all the information added or changed since the last printing of the manual.

Manual updates do not have part numbers. They are numbered sequentially from the time the last edition was issued.

NEW EDITION

A complete revision of a manual; obsoletes all previous editions of the manual and its updates.

A new edition is issued when, due to the scope of the changes involved, it is impractical to issue a manual update.

The date on the title page and back cover of every manual is the printing date of the current edition. This date changes only when a new edition is published. A list of the dates of the manual's previous editions and updates (if any) is kept on the Printing History page of every manual.

Publication of a new edition does not affect the part number of a manual.

If further updates are required, they are made to the new edition. The update numbers run sequentially, starting from the latest edition.

Two COBOL II Manuals

COBOL II/3000 Reference Manual
part number 32233-90001
December, 1979

This reference manual for the newly released COBOL II/3000 compiler has information for the COBOL programmer both on ANSI standard COBOL'74 and on the extended features that COBOL/II 3000 provides. Beginning with standard COBOL constructs and coding conventions, the manual goes on to present and discuss the syntax for all the statements of the four program divisions. Also included are sections on interprogram communication, SORT-MERGE operations, and the COBEDIT program for COBOL copy libraries.

COBOL language principles and syntax are illustrated by a large number of examples interspersed throughout the text. The compiler listing from a sequential update example program is presented in Appendix C along with its symbol table and verb maps and identifier cross-reference. The experienced programmer who needs only to refer quickly to statement syntax will appreciate the convenience of the Composite Language Skeleton in Appendix J.

COBOL/3000 to COBOL II/3000
Conversion Guide (32233-90005)
December, 1980

This new manual helps users convert existing COBOL/3000 programs for execution under COBOL II/3000. Procedures are provided for using the special conversion proram COBCNV. In addition, the manual identifies the differences between COBOL/3000 and COBOL II/3000 (although the two are largely compatible).

A New Manual for IML/3000

IML/3000 Reference Manual
part number 32229-90001
March, 1980

HP's new data communications subsystem, IML/3000, is documented in this manual. IML/3000 permits interactive communications between a host mainframe and remotely-located HP 3000's.

Three HP V/3000 Manuals

HP3000 Data Entry and Forms Management System (V/3000) Reference Manual — 2nd Edition
part number 32209-90001
February, 1980

Using V/3000 — 2nd Edition
part number 32209-90004
February, 1980

Incorporated in these second editions of the HP V/3000 (formerly HP VIEW/3000) manuals are minor changes and corrections as well as the change of the product name. Additionally, explanations of the V/3000 error messages have been added to appendix B of the reference manual.

V/3000 ENTRY Program Operator's Quick Reference Guide — 2nd Edition
part number 32209-90003
February, 1980

Not listed previously in COMMUNICATIOR 3000'S Catalog, the ENTRY Program Operator's Quick Reference Guide is an easy to use summary of the commands and procedures most often used by the data entry operator working with V/3000. The Guide includes information on logging on and off, running ENTRY, entering, correcting, and browsing data, printing forms and data, and recovery both from program termination and power failure. Spiral bound, it will lie flat beside the terminal and provides the user with tabs for ease of reference.

SORT-MERGE/3000 Reference Manual 3rd Edition
part number 32214-90001
March, 1980

This new edition documents major enhancements to SORT-MERGE. SORT-MERGE now supports ASCII or EBCDIC input data and ASCII, EBCDIC, or user defined collating sequences; SORT can now accept more than one input file (thus eliminating the necessity for a MERGE step); and SORT-MERGE will now set the system JCW to FATAL if an alternate output file (OUTPUTnn) is created. In addition, SORT-MERGE performance has been improved through the use of multirecord I/O. If SORT-MERGE is being used programmatically, the files passed to SORT-MERGE via intrinsics may be opened with NOBUF or MR option; SORT-MERGE will perform the buffering and blocking/deblocking.

Series II/III Console Operator's Guide 4th Edition
part number 30000-90013
March, 1980

Series 30/33 Console Operator's Guide 3rd Edition
part number 30070-90025
March, 1980

Both manuals are revised to include the new and enhanced commands available with the 2011 release of MPE. Major areas of revision include Spooler enhancements, new Console Locating and Retrieval commands, and the Foreign Disc Facility.

Index to MPE Documents 4th Edition
part number 30000-90045
January, 1980

This edition updates the Master Index of the MPE 3000 Operating System Manuals through the 1918 release, and adds the index of the Series 33 Console Operator's Guide to the manual.

MPE Commands Reference Manual — Update #2
part number 30000-90009
March, 1980

This update adds two new commands, :REMOTE and :IML. The :REMOTE command allows for communication between a local HP computer and a remote HP computer in a DS/3000 Network environment. The :IML command is a function of the new product IML (Interactive Mainframe Link), an emulator for the IBM 3270 control unit.

MPE Intrinsics Reference Manual — Update #3
part number 30000-90010
March, 1980

This manual has been updated for the 2011 release and includes documentation on Foreign Disc Facility (FDF). FDF allows you to use the file system to access and alter disc packs and flexible diskettes that do not have standard HP3000 file system disc label formats. The addition of FDF does not result in any new intrinsics; however, several existing intrinsics have been modified. These are: FOPEN, FCLOSE, FREAD, FWRITE, FWRITEDIR, FREADDIR, FGETINFO, FFILEINFO, and FCHECK.

MPE System Manager/System Supervisor Reference Manual — Update #2
part number 30000-90014
March, 1980

This update documents the Foreign Disc Facility that permits non-privileged users to manipulate entire disc packs or floppy discs through the file system. In particular, it allows implementation of the IBM 3741 floppy format.

MTS 3000 Multipoint Terminal Software Reference Manual (32193-90002) — Update #1
Feb., 1980

Documents these changes: MPLINE is now a Command Interpreter command; the addition of a switch setting to HP modem table 8-10; and the revision of the software version number.

System Utilities Reference Manual — Update #3
part number 30000-90044
March, 1980

This update corresponds to the 2011 Installation Tape. Optional parameters have been deleted from DPAN2, as have all but one of DPAN2's entry points. Corrections have made to the documentation of DISKED2 and LISTDIR2. System Manager capability is now required to run MEMLOGAN and MEMTIMER. ASOCTABL has a new entry point, LIST. LISTLOG2 output files are assigned to the permanent domain. Several SPOOK commands have been altered.

RJE/3000 (2780/3780) Emulator — Update #2
Reference Manual for Pre-Series II Systems
part number 30130-90001
March 1980

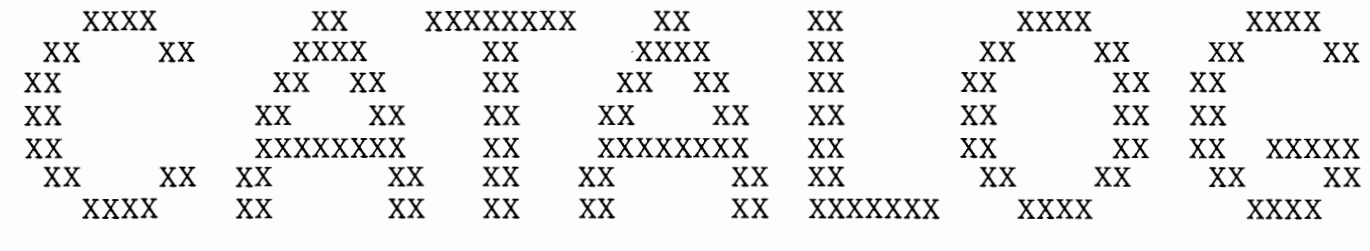
The product name and number have been changed. (The new number is HP 30097A.) Also the new paramters to the RJLINE and RJIN commands are documented.

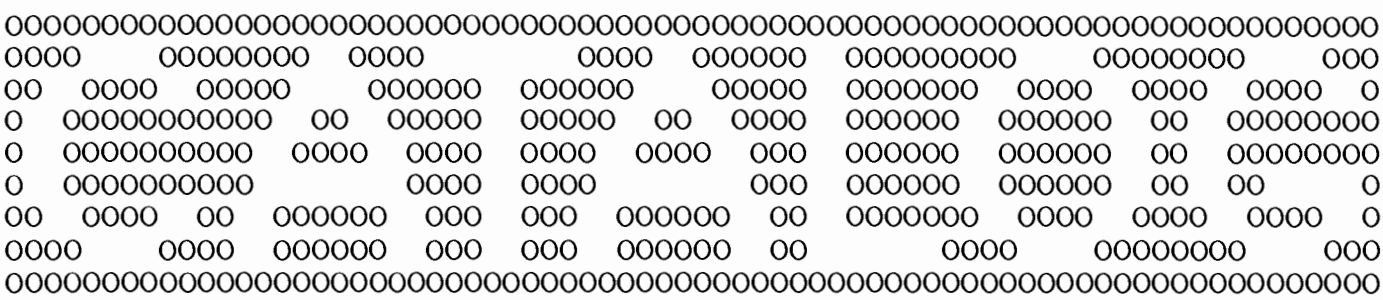
IMAGE Data Base Management — Update #1
Reference Manual
part number 32215-90003
March, 1980

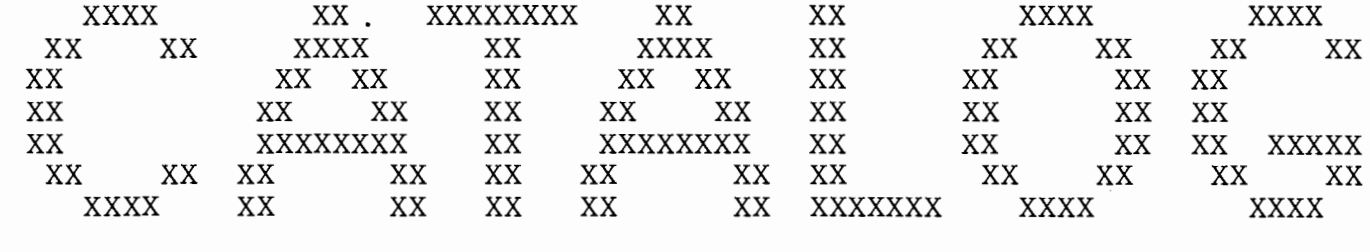
This update documents the following enhancements to IMAGE: two new commands for DBUTIL (SECURE and RELEASE), a dumping option to ENABLE and DISABLE, and an enhanced format in DBUNLOAD for reporting broken chains. Additionally, the following user-suggested topics are discussed and clarified: extended sort fields, DBCLOSE mode 2 vs. mode 3, use of DBLOCK descriptor lists, table overflow, NOABORTS option, DBRECOV, and BASIC interface to IMAGE logging.

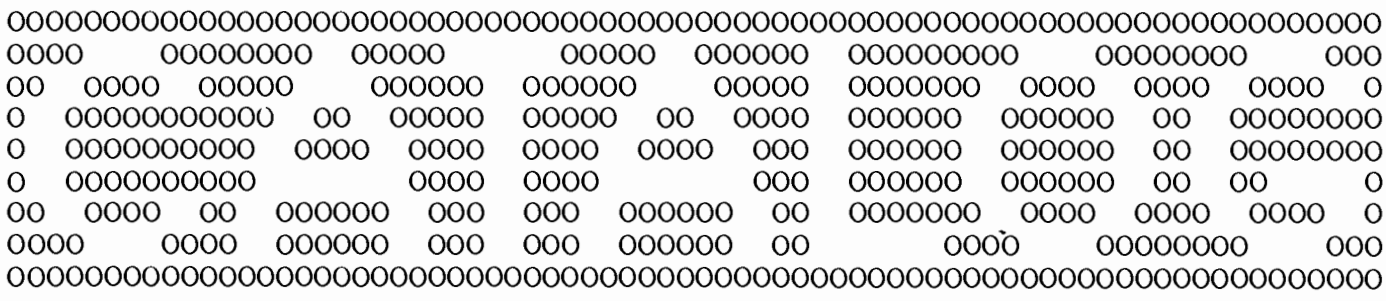
MRP/3000 Material Requirements — Update #1
Planning
User/Administrator's Reference Manual
part number 32388-90001
November, 1979

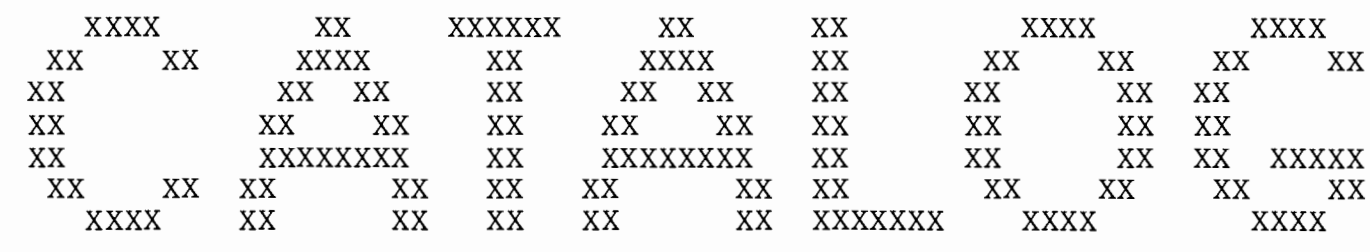
This update documents an enhancement to the MRP Action Report, and adds SPC/3000 to the list of MFG/3000 modules.











KEY

Manuals that are new or have changed since the last edition of this catalog are noted by an asterisk (*) in the leftmost column. An asterisk in the "Price" column indicates that the price of the manual was not available at the time the catalog was printed.

If the V (version) column contains a #, the manual is applicable to systems running MPE III and to those running MPE C. Manuals which apply to MPE C systems only are listed under "MPE C MANUALS".

HP 3000 COMPUTER SYSTEMS

SYSTEM MANUALS

	Manual Title	V	Part Number	Price	Print Date	Up-dated
	Using the HP 3000: An Introduction to Interactive Programming	#	03000-90121	6.50	4/79	
	General Information Manual (Series II/III)		30000-90008	5.25	9/79	
*	MPE Commands Reference Manual		30000-90009	14.75	7/79	3/80
*	MPE Intrinsics Reference Manual		30000-90010	16.50	4/78	3/80
	MPE Segmenter Reference Manual	#	30000-90011	3.50	2/77	
	MPE Debug/Stack Dump Reference Manual	#	30000-90012	4.50	9/76	6/77
	Series II/III Console Operator's Guide		30000-90013	13.50	3/80	

SYSTEM MANUALS (continued)

	Manual Title	V	Part Number	Price	Print Date	Up-dated
*	System Manager/System Supervisor Manual		30000-90014	9.00	7/79	3/80
	Error Messages and Recovery Manual		30000-90015	18.50	6/76	5/78
	HP 3000 Series II/III Machine Instruction Set		30000-90022	5.75	6/76	10/79
*	MPE III System Utilities Reference Manual		30000-90044	4.50	3/77	3/80
*	Index to MPE Reference Documents		30000-90045	4.00	1/80	
	Software Pocket Guide		30000-90049	5.25	4/78	
	Using Files	#	30000-90102	5.25	4/78	
*	Series 30/33 Console Operator's Guide		30070-90025	12.75	3/80	

SUBSYSTEM MANUALS

	Manual Title	V	Part Number	Price	Print Date	Up-dated
	EDIT Reference Manual	#	03000-90012	6.00	8/75	2/79
	Trace Reference Manual	#	03000-90015	4.50	6/76	
	FCOPY Reference Manual	#	03000-90064	4.75	2/78	2/79
	Scientific Library Reference Manual		30000-90027	4.25	6/76	2/77
	Compiler Library Reference Manual		30000-90028	8.50	11/76	
*	SORT Reference Manual	#	32214-90001	3.50	3/80	

LANGUAGE MANUALS

	Manual Title	V	Part Number	Price	†Print Date	Up-dated
	BASIC for Beginners	#	03000-90025	6.00	11/72	
	BASIC/3000 Pocket Guide	#	03000-90050	1.25	9/74	
	System Programming Language Reference Manual	#	30000-90024	9.50	9/76	2/77
	System Programming Language Textbook	#	30000-90025	7.50	6/76	1/77
	BASIC Interpreter Manual		30000-90026	11.00	6/76	8/78
	FORTRAN Reference Manual		30000-90040	8.50	6/76	5/79
	SPL Pocket Guide	#	32100-90001	2.00	11/76	
	FORTRAN Pocket Guide	#	32102-90002	2.50	5/79	
	BASIC Compiler Reference Manual	#	32103-90001	3.00	11/74	6/76
	RPG/3000 Compiler Reference Manual	#	32104-90001	22.00	2/77	
	RPG Listing Analyzer	#	32104-90003	.50	2/77	
	APL Reference Manual		32105-90002	35.00	1/79	
	APL Pocket Guide		32105-90003	4.50	11/76	
	COBOL Reference Manual	#	32213-90001	12.00	7/75	2/79
	Using COBOL: A Guide for the COBOL Programmer	#	32213-90003	6.50	3/78	
*	COBOL/II Reference Mnl.		32233-90001	19.00	12/79	
*	COBOL/3000 to COBOL II /3000 Conversion Guide		32233-90005	3.25	12/79	

DATA COMMUNICATIONS MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
Guidebook to Data Communications	#	5955-1715	3.00	1/77	
RJE/3000 Remote Job Entry (2780/3780 Emulator) Ref. Manual		30000-90047	12.75	11/79	
Data Communications Handbook		30000-90105	10.00	10/78	
HP 30010A Intelligent Network Processor (INP) Installation & Service Manual		30010-90001	4.75	10/79	
HP 30010A/30020A Intelligent Network Processor Diagnostic Procedures Manual		30010-90002	4.25	10/79	
HP 30020A Intelligent Network Processor (INP) Installation & Service Manual		30020-90001	4.50	10/79	
HP 30032B Asynchronous Terminal Controller Instl. & Serv. Manual		30032-90004	14.00	1/74	7/76
HP 30055A Synchronous Single-Line Controller (SSLC) Instl. & Serv. Manual	#	30055-90001	8.50	12/77	4/79
Hardwired Serial Interface (HSI) Instl. & Service Manual		30360-90001	6.00	3/77	5/79

DATA COMMUNICATIONS MANUALS (continued)

	Manual Title	V	Part Number	Price	Print Date	Up-dated
	DS/3000 Reference Manual		32190-90001	19.00	3/77	11/79
	DS/3000 to DS/1000 Reference Manual for HP 3000 Users		32190-90005	7.25	1/78	
	MRJE/3000 Reference Mnl.		32192-90001	8.75	12/79	
*	MTS/3000 Reference Mnl.		32193-90002	6.50	2/80	
*	IML/3000 Reference Mnl.		32229-90001	*	3/80	

MANUFACTURING APPLICATIONS MANUALS

	Manual Title	V	Part Number	Price	Print Date	Up-dated
	EDC/3000 User Reference Manual		32380-90001	20.00	3/78	4/78
	EDC/3000 System Admin. Reference Manual		32380-90002	8.50	3/78	4/78
	IOS/3000 User Reference Manual		32384-90001	25.00	3/78	
	IOS/3000 System Admin. Reference Manual		32384-90002	11.00	3/78	
*	MRP/3000 User-Admin. Reference Manual		32388-90001	19.50	8/78	11/79
	SPC/3000 User Reference Manual		32392-90001	11.00	4/79	

TRANSACTION PROCESSING MANUALS

	Manual Title	V	Part Number	Price	Print Date	Up-dated
	QUERY Reference Manual	#	30000-90042	7.50	6/76	5/79
	KSAM Reference Manual		30000-90079	12.50	5/79	
*	HP V/3000 Ref. Manual		32209-90001	12.75	2/80	
*	HP V/3000 Entry Program		32209-90003	2.50	2/80	
*	Using HP V/3000		32209-90004	8.50	2/80	
*	IMAGE Data Base Management Reference Manual		32215-90003	10.25	9/79	3/80

EDUCATIONAL APPLICATION MANUALS

	Manual Title	V	Part Number	Price	Print Date	Up-dated
	Student Information System Reference Manual	#	32900-90001	13.00	9/74	8/76
	Student Information System Technical Mnl	#	32900-90005	32.00	3/75	
	Student Assignment System Reference Manual	#	32901-90001	15.50	7/75	8/76
	Student Assignment System Technical Manual	#	32901-90005	9.75	8/78	
	College Information System Reference Manual	#	32902-90003	13.00	1/78	
	College Information System Technical Mnl.	#	32902-90005	10.50	2/78	

ADDITIONAL MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
HP 3000 Series System Support Log		03000-90117	20.00	10/78	1/79
HP 3000 CX to HP 3000 Series II Program Conversion Guide		30000-90046	3.50	6/76	
Site Preparation Manual Series II/III		30000-90082	7.00	9/78	
Site Planning Workbook Series II/III		30000-90086	6.00	9/77	
Guide to a Successful Installation	#	30000-90135	7.00	12/79	
Series III(32435A) Site Preparation Manual		30000-90145	2.00	1/79	4/79
Series III(32435A) Site Planning Workbook		30000-90146	5.50	5/79	
Technical Writer's Survival Kit		30000-90171	2.50	7/79	
Series 33 Computer Systems Site Preparation Planning Guide		30070-90007	4.00	10/78	2/79
Series 33 Computer Systems Site Planning Wkb		30070-90029	6.00	9/78	

ADDITIONAL MANUALS (continued)

Manual Title	V	Part Number	Price	Print Date	Up-dated
Series 30 Site Preparation Guide		30080-90002	3.00	8/79	
Series 30 Site Planning Workbook		30080-90003	5.00	8/79	
Series 30 Site Planning Set		30080-60050	6.50	8/79	
HP 2894A Card Reader Punch Operating Manual		30119-90009	11.50	10/76	
Line Printer Operating and Programming Manual		30209-90008	6.75	6/76	
IBM System/3 to HP 3000 Conversion Guide	#	32104-90004	10.75	7/78	

MPE C MANUALS

Manual Title	V	Part Number	Price	Print Date	Up-dated
BASIC Interpreter Reference Manual		03000-90008	9.75	7/75	
Compiler Library Reference Manual		03000-90009	11.50	2/76	
Scientific Library Reference Manual		03000-90010	5.75	7/75	
System Ref.Mnl. Series I		03000-90019	24.00	9/73	3/77
Software Pocket Guide		03000-90126	2.70	7/78	
IMAGE Data Base Management Reference Manual		30000-90041	7.00	12/76	5/78

MPE C MANUALS (continued)

	Manual Title	V	Part Number	Price	Print Date	Up-dated
	MPE Intrinsics Reference Manual		30000-90087	20.00	4/77	4/78
	MPE Commands Ref. Mnl.		30000-90088	20.00	4/77	4/78
	System Manager/System Supervisor Manual		30000-90089	12.50	4/77	4/78
	Console Operator's Guide		30000-90090	11.00	4/77	4/78
	General Information Manual (Series I)		30000-90091	9.25	4/77	
	INDEX/3000 Reference Mnl		30000-90095	10.50	6/77	4/78
	MPE System Utilities Reference Manual		32000-90008	2.05	10/75	
	FORTRAN Reference Manual		32102-90001	10.00	3/76	
*	RJE/3000 (2780/3780 Emulator) Ref. Mnl. for Pre-Series II Systems		30130-90001	9.00	12/74	3/80
	IBM 1130/1800 to HP 3000 FORTRAN Conversion Gd.		36995-90013	4.70	2/75	5/75

NOTES: The printdate of the MRJE/3000 Reference Manual (32192-90001) was incorrectly stated as 1/78 in the last appearance of this Catalog (Issue 23). The correct printdate is 12/79.

The last issue of this Catalog stated that an update was made to the Series II/III Console Operator's Guide (30000-90013) in 12/79. This was incorrect. The manual was not updated between its third edition, dated 7/79, and its current (fourth) edition, dated 3/80.

COBOL II/3000:
Tips on Performance Optimization

This article contains a discussion of performance tips to help a programmer optimize COBOL II/3000 performance.

by Tony Lemberger, General Systems Division

COBOL II/3000 is Hewlett-Packard's high level implementation of the 1974 COBOL ANSI standard. The design and execution of the compiler allow for high performance of applications.

To maximize the performance of COBOL II/3000, the following considerations should be taken into account when writing a COBOL II/3000 program on the HP3000. If the guidelines are followed, the result should be a program that executes more efficiently than one that was programmed without considering these items.

This information pertains to the COBOL II/3000 compiler (version HP32233A.00.01).

1. The compiler allows users to segment both main and subprograms by using section segment numbers. Proper segmentation of a program can have a significant impact on its runtime performance. To illustrate this, an example of how to organize code for efficient segmentation is shown below. Notice that seldom-used logic (such as data base opening and closing operations) is placed in an initialization segment, and main processing takes place in one or more other segments.

```
 INIT SECTION 10.
 DB-OPEN
*  DATA BASE OPEN LOGIC.
 DB-CLOSE.
*  DATA BASE CLOSE LOGIC.
 OTHER-INIT.
*  OTHER INITIALIZATION LOGIC.
 BEGIN-PROC SECTION 20.
*  BEGIN MAINLINE PROCESSING.
*  END OF EXAMPLE
```

It is very important to minimize the crossing of code segment boundaries with PERFORM's, GO TO's or CALL's. A program should exhibit a great degree of locality, meaning that once it starts execution in a particular code segment, it should remain in that code segment as long as possible.

2. Reduce the global areas in stacks. The only information that should be in global areas (DB-relative main program storage) is information that must be accessible to more than one procedure or subprogram. This can be done by creating all subprograms as dynamic subprograms, which will cause these subprograms to use Q-relative local storage rather than DB-relative global storage, and, thus, significantly reduce stack requirements. In addition, static information such as error messages, display messages, and numeric constants should be coded in-line in the program as literals enclosed in quotes, or in the case of numeric constants, as explicit constants in the source. That will place those items at PB-relative addresses, which means they will actually be part of the code segment. This, again, will significantly reduce stack sizes. An even greater gain can be seen if there are multiple users running the same program file. Instead of having a separate copy of constants and error messages in each user's stack, there will be one copy located in a sharable code segment. For example:

```
*   TO REDUCE GLOBAL AREAS IN STACKS, PUT ERROR
*   MESSAGES IN PB-RELATIVE ARRAYS.  IN ADDITION,
*   IF THE APPLICATION ALLOWS, PUT ERROR HANDLING
*   IN ITS OWN CODE SEGMENT.
*
 ERROR-ROUTINE SECTION 30.
      GO TO ERR-1, ERR-2 DEPENDING ON ERROR-NUMBER.
 ERR-1.
      MOVE "INVALID FILE NUMBER" TO ERR-MESS.
      PERFORM ERROR-PRINT.
 ERR-2.
      MOVE "END OF FILE" TO ERR-MESS.
      PERFORM ERROR-PRINT.
*
*   ANY OTHER ERROR HANDLING SHOULD GO HERE IF
*   POSSIBLE
*
 CONTINUE-ON SECTION 40.
*
*   THIS WILL BEGIN A NEW CODE SEGMENT FOR OTHER
*   PARTS OF THE PROGRAM
*
*   END OF EXAMPLE
```

3. Use COMP for numeric data items of <=9 digits, and COMP-3 for > 9 digits. The compiler sets up integer storage for COMP items of 1 to 4 words in size, and uses packed decimal storage for COMP-3 items. Integer numbers can be stored in less space, and can be operated upon rather quickly (as compared to packed decimal numbers which require more storage space and longer execution times to manipulate).

4. Use signed instead of unsigned data. Having to compute the absolute value of a result after it is obtained will thereby be avoided. This affects COMP-3 and DISPLAY items more than COMP and can result in a moderate savings in execution time.

Performance Checklist

Here is a summary of this article:

* Reduce the global areas in all stacks. The only information that should be in global areas is information that must be accessible to all procedures or subroutines, or information that must be modified by more than one procedure. Error messages should, for example, reside in code segments. Make data local to procedures if possible.

* Minimize crossing of code segment boundaries during the execution of a program. This means that efficient segmentation of a program will cause the program to stay in one code segment as long as possible.

* Users can effectively segment their programs and subprograms through the use of section numbers.

* Use COMP-3 for data items > 9 digits, and COMP for <= 9 digits.

* Use signed rather than unsigned data.

****

COBOL II/3000, HP's high level implementaion of the 1974 ANSI standard for COBOL, is designed to be an efficient compiler for both application development and run-time performance. The considerations mentioned in this article can help to significantly increase the performance of programs written using COBOL II/3000.

SORT–MERGE/3000 Enhancements

by Carol Agne, General Systems Division

A new version of SORT-MERGE/3000 is now available. This article will give examples of some of the enhancements, including multiple input files, alternate collating sequences and new SORT-MERGE intrinsics. Consult the SORT-MERGE note files in this issue, and the new SORT-MERGE/3000 Reference Manual, for more information on these enhancements.

Multiple Input Files

Multiple input files are treated by SORT as one contiguous file in which the characteristics of the first file specified are used for all files (input and output). The following example shows an interactive specification for multiple input files:

```
>INPUT (File1,File2,File3)
```

Since input files don't have to be individually sorted and then merged to the output file, only half as many input-output operations are now required.

Assume 3 unsorted input files - File1, File2, and File 3 - each containing 10 records:

WITH SEPARATE OPERATIONS:	Reads	Writes
SORT FILE1 creating SF1	10	10
SORT FILE2 creating SF2	10	10
SORT FILE3 creating SF3	10	10
MERGE SF1, SF2, SF3 creating SF123	30	30
	TOTAL of 120 I/O's	

WITH MULTIPLE INPUT FILES:		
SORT FILE1, FILE2, FILE3 creating SF123	30	30
	TOTAL of 60 I/O's	

With the advent of multiple input files, the task is reduced from N+1 steps to only one step, where N is the number of files to be sorted and then merged. All of the files can be sorted and merged by running SORT.PUB.SYS only once.

Alternate Collating Sequence

ASCII input data can now be sorted or merged in ASCII sequence, EBCDIC sequence, or either sequence further modified ("user-defined"). EBCDIC data can be accepted and can be sorted or merged in either EBCDIC or ASCII sequence.

For instance, assume a user has ASCII data and desires a basic collating sequence of ASCII modified so that letters will be sorted strictly alphabetically, regardless of whether the letter is upper or lower case. The example on the following two pages demonstrates one way the user can achieve the desired effect.

```
:EDITOR
HP32201A.7.05 EDIT/3000  WED, FEB 20, 1980,  3:22 PM
(C) HEWLETT-PACKARD CO. 1979
/t outside
FILE UNNUMBERED
/l all
     1      OAK
     2      Radish
     3      bush
     4      Birch
     5      oleander
     6      fir
     7      ivy
     8      Dogwood
     9      eucalyptus
    10      EVERGREEN
    11      daisy
    12      lemon
    13      tulips
    14      redwood
    15      ROSE
/e

END OF SUBSYSTEM
:run sort.pub.sys

HP32214B.02.03  SORT/3000 WED, FEB 20, 1980,  3:23 PM
(C) HEWLETT-PACKARD CO. 1980

>input outside
>data ascii, seq ascii
>show sequence
nul soh stx etx eot enq ack bel  bs  ht  lf  vt  ff  cr  so  si
dle dc1 dc2 dc3 dc4 nak syn etb can  em sub esc  fs  gs  rs  us
 sp   !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /
  0   1   2   3   4   5   6   7   8   9   :   :   <   =   >   ?
  @   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O
  P   Q   R   S   T   U   V   W   X   Y   Z   [   \   ]   ^   _
  `   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o
  p   q   r   s   t   u   v   w   x   y   z   {   |   }   ~ del

>altseq "A-Z" = "a-z"
nul soh stx etx eot enq ack bel  bs  ht  lf  vt  ff  cr  so  si
dle dc1 dc2 dc3 dc4 nak syn etb can  em sub esc  fs  gs  rs  us
 sp   !   "   #   $   %   &   '   (   )   *   +   ,   -   .   /
  0   1   2   3   4   5   6   7   8   9   :   :   <   =   >   ?
  @   [   \   ]   ^   _   `  A=   a  B=   b  C=   c  D=   d  E=
  e  F=   f  G=   g  H=   h  I=   i  J=   j  K=   k  L=   l  M=
  m  N=   n  O=   o  P=   p  Q=   q  R=   r  S=   s  T=   t  U=
  u  V=   v  W=   w  X=   x  Y=   y  Z=   z   {   |   }   ~ del
```

Figure 1. Example of Collating Sequence Using SORT-MERGE/3000.

```
>key 1,10
>output showsort
>end
                        STATISTICS

NUMBER OF RECORDS =                                  15
NUMBER OF INTERMEDIATE PASSES =                       0
SPACE AVAILABLE (IN WORDS) =                     13,748
NUMBER OF COMPARES =                                 58
NUMBER OF SCRATCHFILE IO'S =                         12
CPU TIME (MINUTES) =                                .00
ELAPSED TIME (MINUTES) =                            .02
RECORD SIZE (IN BYTES) =                             72
SCRATCH FILE SIZE (# SECTORS) =                     105

END OF PROGRAM
:editor
HP32201A.7.05 EDIT/3000  WED, FEB 20, 1980,  3:25 PM
(C) HEWLETT-PACKARD CO. 1979


/t showsort
FILE UNNUMBERED
/l all
     1      Birch
     2      bush
     3      daisy
     4      Dogwood
     5      eucalyptus
     6      EVERGREEN
     7      fir
     8      ivy
     9      lemon
    10      OAK
    11      oleander
    12      Radish
    13      redwood
    14      ROSE
    15      tulips
/e

END OF SUBSYSTEM
```

Figure 1. Example of Collating Sequence Using SORT-MERGE/3000. (Continued)

New SORT-MERGE Intrinsics

SORTINIT must be used in place of SORTINITIAL or SORTINITIALF if multiple input files or an alternate collating sequence is desired.

New MERGE intrinsics (MERGEINIT, MERGEOUTPUT, and MERGEEND) have been implemented which are similar to the SORT intrinsics.

Other Enhancements to SORT-MERGE/3000

* Performance of the SORT and MERGE programs has been improved through the use of unbuffered I/O on the input and output files.

* If an OUTPUTnn file is created, the JCW is set to FATAL.

* COBOLII-related enhancements have been added. (See the COBOLII Reference Manual for details.)

* The EXIT command allows the user to terminate SORT-MERGE without performing a sort or merge.

* SORT-MERGE supports programmatic execution of most MPE commands.

* Most commands and key types can be abbreviated.

* The control-Y trap facility to track the progress of SORT now prints the numbers of records merged during the intermediate passes.

To Issue 23:

On page 12, in the article "IMAGE Logging and Recovery: Questions and Answers", it was stated that "Reaching the end of disc logfile is therefore similar in effect to a system failure and should be avoided." This is not true, as a system failure is a much more serious condition. The correct statement is: "Overflowing the logfile should be carefully avoided since it requires a complete database recovery to remove any incomplete transactions."

To Issue 22:

On page 24, in the article "Use of STACK= and MAXDATA=", in reference to the PCBX (Process Control Block Extension), it was stated that "This information includes directory pointers, the DSTs for any allocated extra data segments, and file control blocks (FCBs). This area of the stack is inaccessible by the normal user and is initially about 1200 words long." This is incorrect. The corrected statement is: "This information includes directory pointers, the DSTs for any allocated extra data segments, and file control information. This area of the stack is inaccessible by the normal user and is initially 258 words long."

On page 25 of the same article, the size of the overflow area was incorrectly stated as 128 words. The correction is that this number may vary.

Also on page 25 of this article, the statement "The file system uses the PCBX for file control blocks; if many files are to be opened concurrently, the area may need to be expanded to accommodate more FCBs" was incorrect. The correction is: "The file system uses the PCBX for file control information; if many files are to be opened concurrently, the area may need to be expanded to accommodate a larger available file table."

The following information should be added to the article: "Error 71 is returned by the file system when an available file entry cannot be obtained. This can occur either because a) there are already 255 files open (the file system maximum), or, b) because the available file table cannot be expanded to accommodate a new entry."