VOLUME 2
ISSUE 1

HEWLETT
PACKARD

COMMUNICATOR 3000

HP 3000 Computer Systems

COMMUNICATOR 3000

HEWLETT
PACKARD

19447 PRUNERIDGE AVENUE, CUPERTINO, CA 95014

Part No. 5955-1770
Volume 2, Issue 1

Printed in U.S.A. 7/84

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

CONVENTIONS USED IN THIS MANUAL

NOTATION — DESCRIPTION

COMMAND

Commands are shown in CAPITAL LETTERS. The names must contain no blanks and be delimited by a non-alphabetic character (usually a blank).

KEYWORDS

Literal keywords, which are entered optionally but exactly as specified, appear in CAPITAL LETTERS.

parameter

Required parameters, for which you must substitute a value, appear in ***bold italics***.

parameter

Optional parameters, for which you may substitute a value, appear in *standard italics*.

[ ]

An element inside brackets is optional. Several elements stacked inside a pair of brackets means the user may select any one or none of these elements.

Example:
```
[ A ]
[ B ]   user may select A or B or neither.
```

When brackets are nested, parameters in inner brackets can only be specified if parameters in outer brackets or comma place-holders are specified.

Example: [*parm1*[,***parm2***[,***parm3***]]] may be entered as:

parm1,***parm2***,***parm3*** or
parm1,,***parm3*** or
,,***parm3*** , etc.

{ }

When several elements are stacked within braces the user ***must*** select one of these elements.

Example:
```
{ A }
{ B }   user must select A or B.
```

...

An ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted.

user input

In examples of interactive dialog, user input is underlined.
Example: NEW NAME? ALPHA1

superscript^C

Control characters are indicated by a superscript^C. Example: Y^C. (Press Y and the CNTL key simultaneously.)

RETURN

RETURN indicates the carriage return key.

<<COMMENT>>

Programmer's comments in listings appear within << >> .

** Comment **

Editor's comments appear in this form.

CONTENTS

SECTION/ARTICLE PAGE

INTRODUCTION

COMMANDS

INTRINSICS

UTILITIES

SYSTEM SOFTWARE ENHANCEMENTS

CONTENTS (continued)

SECTION/ARTICLE PAGE

DATA COMM ENHANCEMENTS

NEW PERIPHERAL SUPPORT

NON-MPE SPECIFIC

PREFACE

We at HP are eager to offer you the latest software version of MPE, called MPE V/E (G.00.00). This software release, by utilizing the MPE table expansions and a wide range of other improvements, provides a major upgrade for your system. We feel that you will be pleased with our efforts at tailoring MPE V/E to better meet your information needs.

The Communicator 3000 is designed to provide you with new system documentation between manual updates. The articles in this manual highlight specific MPE changes that occurred since the C/D.01.00 ("Q-MIT") version, released in December 1982. For easy referencing, the articles have been arranged into eight different sections.

Leading off the manual, in the introductory section, is an article that references the most important material covered in this book. Also included in this section is an article on MPE V/E table expansions and disc caching, an article on MPE V/E compatibility issues, and one on the documentation plan for MPE V.

The next three sections specify commands, intrinsics, and utilities that are either new or enhanced. Among the articles in these sections is one that explains the new disc caching commands, one on :STORE enhancements, one on :RESTORE enhancements, an article on the new intrinsic JOBINFO, and one on MPE utility program modifications.

The fifth section, System Software Enhancements, is of particular interest to our users. This section begins with an article that discusses the firmware upgrade for all CS-80 products supported on the HP 3000. Section V also includes articles relating to the table and directory expansions, articles discussing enhancements to data base management and performance measurement products, and an article specifying the new MPE stack size requirements for file system intrinsics, among others.

The sixth section talks about enhancements related to data communications, with articles on DS, MTS, a new product called the Workstation Configurator, and the new ADCC software driver code.

The seventh section, which observes the new peripherals supported on MPE V/E, includes articles on the HP 7974A and 7978A tape drives, the HP 2687A/2688A and 2563A printers, and the 1/4-inch tape driver change.

The final section brings together the various subsystems that are available with MPE V/E. These articles include enhancements to SPL/3000, Pascal/3000, Dictionary/3000, and VPLUS, suggestions of ways to help you to.improve the performance of HPWORD, graphics support considerations, and many others.

We hope that you enjoy reading this manual.

Larry S. Lodovisi, Editor

INTRODUCTION

SECTION
1

Articles for Special Consideration

By Donna Sidman

There have been many changes made in the release of MPE V/E (G.00.00), and some of these changes may concern your present applications. Prior to updating to MPE V/E, you should read this entire package. To ensure that you review the most important material, pay special attention to the following topics:

MPE V/E Offers Increased Performance and Expandability

By Jeff Byrne and Larry Russell

The G.00.00 release of MPE V/E adds major enhancements to the capacity and performance of HP 3000 systems. MPE V/E enables users to take advantage of the expanded tables, which provide the software support for a greater number of users, applications, and devices. Another major enhancement is disc caching, which significantly improves the response time and transaction throughput of I/O-intensive applications. (Disc caching is available only on a High Performance Series 39, and on Series 42, 48, and 68 systems.)

MPE V/E Table Expansions

Most of the major tables, including the Data Segment Table (DST) and Process Control Block Table (PCB), are being expanded fourfold to provide room for future expandability. In addition, the Code Segment Table (CST) and Extended Code Segment Table (CSTX) have been significantly enlarged. These extensions offer greater capacity for HP segmented library segments, as well as user program segments. These changes also allow user applications to be segmented into smaller sections for more efficient memory management, and increase the maximum program size from 2 to 8 Mb.

Users who want to take full advantage of the table expansions will require MPE V/E microcode. This firmware is available as an upgrade product for installed Series 39, 40, and 44, and is standard on the High Performance Series 39, and on Series 42 and 48. (Because Series 64 and 68 use Writable Control Storage to load the system microcode, a firmware upgrade is not required to obtain the full benefit of the expanded tables.)

With the new microcode, the following table maximums will be in effect:

TABLE	OLD MAX (entries)	NEW MAX (entries)
DST	1024	4096
PCB	256	1024
DRQ	256	900
IOQ	256	1300
SWAP TABLE	2048	5400
ICS	2048	4096
UCOP REQ QUEUE	256	1024
TRL	128	1024
USERS/LOG PROCESS	128	256
LOGICAL DEVICES	255	999
OPEN SPOOLFILES	255	1024
CST	192	2048
SEGMENTS/PROGRAM	63	255

NOTE

The expansions to the CST and CSTX are not available without the MPE V/E firmware.

Although individual tables may be expanded to the maximum sizes indicated above, the number of tables which may be expanded concurrently without the firmware upgrade is limited by space in bank zero. Customers who need enlarged CST/CSTX tables, and who want to configure system tables to their maximum limits should purchase the enhanced firmware as either a product or system upgrade.

The larger Logical Device Table may enable you to significantly increase the number of terminals and peripherals configured on your system. The enlarged PCB table permits more users to log on concurrently, allowing up to 400 simultaneous sessions on the Series 68. Finally, the extensions to the Code Segment Table (CST) and Data Segment Table (DST) can provide users with the capability to run a greater number of larger programs on MPE V/E, each having access to a larger amount of data.

Disc Caching

This innovative upgrade product available on the High Performance Series 39, and on Series 42, 48, and 68 with MPE V/E significantly improves system throughput on HP 3000s running in I/O- and database-intensive environments. Disc caching enhances system performance by using main memory as a high speed buffer for disc. Disc caching places those most frequently accessed portions of files and directories in available regions of main memory where they can be accessed repeatedly by an executing program. Thus, instead of going immediately to the disc to get requested information, disc caching first checks to see if the file information is already in main memory (in the disc cache). When the information is located in the disc cache, I/O performance is improved in two principal ways. First, several time-consuming disc accesses are eliminated. Second, information may be accessed in main memory 10-100 times faster than it may be accessed from disc; disc caching greatly reduces the time required for an I/O transaction to complete. This improvement means better response and higher throughput for I/O-intensive applications.

Disc caching is most effective in applications in which multiple users share the same information. Programs which share data bases, such as IMAGE/3000, benefit greatly. Disc caching not only reduces the contention for these data bases it also reduces the lock service times of the database control blocks because I/O requests can be serviced more quickly.

The MPE Memory Manager controls the operation of disc caching, and handles file information in the cache dynamically and transparently, just as a data segment would be treated.

As a result, special areas of main memory need not be dedicated for the disc cache. However, users who wish to gain significant performance benefits from disc caching must configure an adequate amount of main memory on their systems. Your local SE or performance specialist can help you decide how much additional memory will be required, if any.

Although the operation of disc caching is fully transparent to the user, the System Manager or System Supervisor can enable, disable, and control caching through designated commands. Since disc caching can be enabled on a device-by-device basis, only those disc drives which will benefit from caching need to have caching enabled. Full guidelines on the use of all the disc caching commands are provided in the article entitled "DISC CACHING COMMANDS".

Other Major Features of MPE V/E

MPE V/E contains several other new features in addition to the table expansions. The increased functionality of :RESTORE and the enhanced version of :STORE provide HP 3000 users with better error tolerance and faster file backup. Modifications have been made to I/O drivers to support Rotational Position Sensing (RPS), which allows significantly more I/Os to be performed when multiple CS-80 drives are connected to the same General Interface Channel (GIC).

Along with these performance features, several new commands and intrinsics, as well as some enhancements to utilities, subsystems, languages and applications are included in this release. All these items are detailed elsewhere in this issue of the MPE V/E Special Documentation Package.

MPE Compatibility

HP 3000 systems running MPE IV or MPE V/P will be upward compatible with all systems running MPE V/E, with or without the firmware upgrade. Thus, we expect that all user applications or programs developed on MPE IV or MPE V/P will execute without modification on an MPE V/E-based system.

Applications written on an MPE V/E system must not exceed the MPE IV table limits in order to run on MPE IV- or MPE V/P-based systems. Similarly, programs developed on MPE V/E with updated firmware must not exceed the newly configured table limits on MPE V/E systems without firmware.

Compatibility Issues

By Donna Sidman

Because of the major changes made to MPE in the release of MPE V/E (G.00.00), there are several compatibility issues of concern:

- Programs developed on MPE V/E software with the new microcode might not run on a system without the new microcode, due to the increase in code segments that MPE V/E allows in a program. The 63 code-segment-per-program limitation has been eliminated in MPE V/E, assuming the new microcode is installed. (Refer to the "CODE SEGMENT TABLE EXPANSION IN MPE V/E" article for more details.)

- Privileged Mode programs/applications may require modification to run on MPE V/E, in the case where the program/application modifies or accesses tables that have changed with MPE V/E. (Please note that the tables can be expanded to a specified limit regardless of the microcode.) Programs that expect a table to be within Bank 0 will also have to be modified to accommodate the deletion of this limitation, in MPE V/E with the new microcode.

CAUTION

> It is important to eliminate any programs with PM capability that currently exist on your MPE IV or MPE V/P system (not yet modified for MPE V/E changes) before doing an update. On most systems, this will require a file-by-file search through any account that has PM capability, looking for programs that were prepared with PM. (Consult your SE if you need help recognizing any PM application that might be running on your system.)
>
> An additional warning: If unmodified PM programs are run under MPE V/E, various symptoms can occur, such as System Failure 16, System Failure 249, System Failure 10, System Failure 623, and program aborts.

- Assembly code may need to be modified to run successfully on MPE V/E with the new microcode. This is due to the PLABEL definition change, which eliminates the internal label case of PCAL 0. (Consult your SE for more details.)

- In reference to the above compatibility issue, some BASIC compiled programs may require recompilation, as the BASIC compiler on MPE IV-based systems may generate this illegal PCAL 0 instruction sequence. A utility called PVERIFY.PUB.SYS is available on MPE V/E and can be used to search through program files to identify any location of the illegal construct. (Refer to the "IMPACT OF CST EXPANSION ON COMPILED BASIC PROGRAMS" article for more details.)

- Programs that check for PARM and INFO parameters by tracing back to the Morgue stack marker will have to modify their scheme on MPE V/E with the new microcode. Traditionally, looking for a delta-P value of 0 in a stack marker was a common mechanism for finding the Morgue stack marker, and then locating PARM and INFO. But with MPE V/E and the new microcode, the delta-P in the Morgue stack marker has the value of %40000, rather than 0, because of the physical-mapping bit in the new MPE V/E stack marker.

- Existing applications that use the FFILEINFO or FGETINFO intrinsics may have to be modified if they are to run on a system with devices having logical device numbers greater that 255. Two new items (50,51) have been added to these intrinsics to accommodate the larger logical device numbers. (Refer to the "FFILEINFO: 16 BIT LDEVs AND OTHER ENHANCEMENTS" article for more details.)

- There are some MPE IV-based utilities that will not run on a MPE V/E system (with or without the new microcode) due to modifications to their method of table access. (Refer to the "MPE V/E UTILITIES" article for more details.)

- MPE IV-based system software, which includes MPE V/P, wiil not run on the new MPE V/E microcode.

- MPE V/E system software will run on both the current and new sets of microcode.

- MPE V/E, without the new microcode, might not successfully update from an MPE IV system with a large I/O configuration. This is due to the entry size increase in many of the tables, and it is probable that there will be little room in Bank 0 Memory.

- As with each new release of MPE, the stack size requirements for file system intrinsics have increased with MPE V/E. Therefore, some programs that ran correctly on a pre-MPE V/E system might now run out of stack space on MPE V/E. (Refer to the "MPE STACK SIZE REQUIREMENTS" article for more details.)

The PCONVERT Conversion Program

By Donna Sidman

Before updating a system from C/D.00.02, C/D.01.00, C/D.01.01, C/D.01.02, or E/F.00.00 (MPE V/P) to G.00.00 (MPE V/E), it is necessary to run the conversion program PCONVERT. This program converts the following tables that have changed with MPE V/E:

I/O Tables

CS Table
CS Def Table
CS Additional Driver Table
Driver Table
Device Class Table
Terminal Type Descriptor Table
LPDT
LDT
LDTX
Device Class Table Header

Other Tables

RIN Table
CONFDATA File
Cold Load Information Table
SYSDUMP/INITIAL Communication Table

PCONVERT must be run before performing the update to MPE V/E, or before performing a backward conversion from MPE V/E to a pre-MPE V/E system. Execution requires SM capability, and the logon must be MANAGER.SYS. Enter the following to run the program:

```
:RUN PCONVERT
```

PCONVERT will automatically determine whether or not a forward or backward conversion is required. It will attempt a forward conversion (from an MPE IV-based table structure to an MPE V/E table structure) if executed on an MPE IV-based system; it will attempt a backward conversion (from the expanded table structure to the MPE IV-based table structure) if executed on a MPE V/E system.

This program will also change the following driver names for you:

In the forward conversion

HIOTERM0 Type 16 --> HIOTERM2
HIOTERM0 Type 32 --> HIOASLP2

In the backward conversion

HIOTERM2 --> HIOTERM0 Type 16
HIOASLP2 --> HIOTERM0 Type 32

PCONVERT first goes through a verification process, checking to see if the table conversion can be performed, and then it performs the actual table conversion.

The verification portion of the program goes through the following checks:

- It will check if this CPU model will support MPE V/E.

- It will check to see if there is enough main memory to support MPE V/E.
- It will check to see if there is enough disc space to update to MPE V/E. (The requirement is 14,000 sectors for the system SL; 2,000 sectors for the system programs; and 1,000 sectors for the newly converted tables.)
- It will check for any nonstandard drivers that might be configured. These devices should be deleted from the configuration before the update.
- It will check that core-resident tables will fit in main memory. CS products are not included in this calculation.
- On a backward conversion, it will ensure that no tables are over the MPE IV-based maximums.
- On a backward conversion, it will check to see if the directory has been expanded; if so, it will indicate that an update must be performed on a tape with the directory patches installed.

The verification process does not check for enough Bank 0 Memory. Because Bank 0 availability is not checked by PCONVERT, running out of Bank 0 on the update to MPE V/E can occur.

Following is a list of error messages that may appear if one of the verification checks fail:

-
  ```
  WARNING -- This system does not have enough disc space on
     LDEV 1 to convert to MPE V/E.
     You need an additional "XXXXX" sectors of contiguous disc space.
  ```

-
  ```
  WARNING -- NON-STANDARD DRIVER ON THE FOLLOWING LDEV(S)
  1 4 (ldevs 1 and 4)
     I/O table use not calculated for the above ldev(s)
     Non-standard drivers should be deconfigured before an UPDATE
     is attempted.
  ```

-
  ```
  WARNING -- THE FOLLOWING TABLES ARE OVER PRE-MPEV/E MAXIMUMS

  TABLE NAME             CURRENT SIZE             PRE-MPEV/E MAX

  (A list of the appropriate tables and sizes will exist here)
  ```

-
  ```
  WARNING -- DIRECTORY IS GREATER THAN 6112 SECTORS
     You must UPDATE with one of the following MITS
        MPEV/P (E.B0.00)
        MPEV/P WITH DIRECTORY PATCHES
        Q-MIT WITH DIRECTORY PATCHES
        CONSULT WITH YOUR LOCAL SE
  ```

If any of these verification steps fail, no conversion will be performed. An appropriate error message will be printed to inform the user as to why the verification failed.

The system has to be up to run PCONVERT. The program must be run on a stand-alone system; once the tables conversion is complete, it halts the system.

In addition to doing the tables conversion, PCONVERT also purges the MPE IV-based utilities so that they can be replaced with the MPE V/E utilities. There is one exception: IOCDPN0 is not purged by

PCONVERT. The MPE V/E version of IOCDPN0 will be automatically installed on the update to MPE V/E.

If doing a forward conversion and you run out of Bank 0 Memory, you may reduce the table sizes during the MPE Initiator dialog by answering YES to "SYSTEM TABLE CHANGES?".

NOTE

Having the capability of changing the system table sizes on a COOLSTART is an enhancement with MPE V/E. All of those tables listed under "SYSTEM TABLE CHANGES?" in the INITIAL and SYSDUMP dialogs are candidates for changes on a COOLSTART, with the exception of User Logging.

If doing a backward conversion, you cannot reduce memory (without deconfiguring devices), so it is important that customers have an MPE IV cold load tape from their previous systems before the update.

Make sure that all table configurations are returned to a value below or equal to the MPE IV-based maximums. This can be accomplished by performing a cold load from a previous MPE IV-based cold load tape.

MPE V/E Documentation Has a New Look

By Brian B. Egan

In keeping with the spirit of innovation present in the release of MPE V/E, the MPE Documentation Group is proud to announce a new manual set for MPE V. The manuals are in production now, and will be distributed as soon as possible. The figure on the next page gives an overview of the manuals in the set.

The Commands, Intrinsics, and Utilities manuals, and the Communicator, have all been completely revised for MPE V, and have been given new part numbers. (The MPE IV versions will continue to be available.)

In response to reader surveys and comments, the System Manager/System Supervisor manual and the Console Operator's Guide have been combined into an all-new manual, the MPE V System Operation and Resource Management Reference Manual (32033-90005). The new manual has been extensively rewritten to clarify old concepts and to add new material.

The Guide for the New User (32033-90009), written in a friendly, easy to use format, is intended to introduce the HP 3000 to those who have had no previous experience.

We are also currently producing a new "Pocket Guide": the MPE Quick Reference Guide, intended for publication this year. This book will cover both MPE IV and MPE V, and will be produced in a handy, easily updated format.

Finally, two MPE IV manuals have been obsoleted for MPE V. The Index to MPE Documentation will be replaced by a comprehensive second index in every MPE V manual. The System Error Message and Recovery Manual will be replaced by enhancements to the system message catalog and HELP facility, presenting the user with recovery information on the screen at the same time that an error message occurs.

The Segmenter, Debug/Stack Dump, and File System manuals will continue in their present forms as joint MPE IV/MPE V documents. They will be revised as soon as possible to include the latest information.

There are many more manuals applicable to the HP 3000. A complete list may be found in every issue of the MPE V Communicator. Please ask your System Manager.

The MPE Documentation Group is committed to providing you with the most accurate, timely and usable documentation possible. We welcome your comments and suggestions; please submit them via the postpaid Reader Comment Sheets found at the back of each manual, or by a Service Request (SR).

MPE V MANUAL PLAN

INTRODUCTORY LEVEL:

GENERAL
INFORMATION
Manual
5953-7553

GUIDE FOR THE
NEW USER
32033-90009
IN PROGRESS

GUIDE FOR THE
NEW OPERATOR
32033-90021
IN PROGRESS

STANDARD USER LEVEL:

MPE V COMMANDS
Reference
Manual
32033-90006

MPE V INTRINSICS
Reference
Manual
32033-90007

MPE V UTILITIES
Reference
Manual
32033-90008

SEGMENTER
Reference
Manual
30000-90011

DEBUG/STACK DUMP
Reference
Manual
30000-90012

FILE SYSTEM
Reference
Manual
30000-90236

ADMINISTRATIVE LEVEL:

MPE V SYSTEM OPERATION
& RESOURCE MANAGEMENT
Reference Manual
32033-90005

SUMMARY LEVEL:

MPE QUICK
REFERENCE GUIDE
30000-90049
IN PROGRESS

MPE Reports Its Base V.UU.FF

By Jon Cohen

By using the :SYSDUMP command, any user with System Manager (SM) or System Supervisor (OP) capabilities can generate a modified version of the MPE system tape. Usually, this system tape is simply a backup of the system. Occasionally, users build their own versions of the system software with :SYSDUMP. For example, a "patched" system tape is often generated with :SYSDUMP.

One of the modifiable aspects of MPE is the Version, Update, and Fix number associated with the system software. It is reported in the form *V.UU.FF* when the user logs on, and with the :SHOWME command.

However, changing the *V.UU.FF* can cause a problem for HP support personnel. If a user reports a problem on a non-standard *V.UU.FF*, and the problem is sent to the factory for resolution, we have no idea which version of the software to examine. With this new software release, MPE V/E now reports both the user-defined *V.UU.FF* number and the official base *V.UU.FF* of the HP version of the operating system software. (Note that the official *V.UU.FF* is not modifiable by a :SYSDUMP .) The base *V.UU.FF* will agree with the version number on the Printing History page of the applicable MPE Manual.

Starting with this software release, the logon banner looks like:

```
HP3000 / MPE V.  X.YY.ZZ (BASE G.00.00) TUE, NOV 1,...
```

where *X* is the user-defined version, *YY* is the user-defined update number, and *ZZ* is the user-defined fix number. In the sample output of the :SHOWME command below, G.00.00 is the base *V.UU.FF* :

```
USER: #S1269 JON.OSE,COHEN     (IN PROGRAM)
MPE VERSION: HP32033X.YY.ZZ.    (BASE G.00.00)
CURRENT: THU, OCT 27, 1983,  4:54 PM
LOGON:   THU, OCT 27, 1983,  8:53 PM
CPU SECONDS: 9          CONNECT MINUTES: 482
$STDIN LDEV: 68          $STDLIST LDEV: 68
```

COMMANDS

SECTION

II

Differences in Command Classifications; Two New MPE Capabilities

By Jon Cohen

In the past, users have been confused between "OP" Commands, Operator Commands, and Console Commands. This article will attempt to clarify the differences between these definitions. The two new MPE capabilities are also included.

Operator Commands, ”OP” Commands, and Console Commands

An Operator Command typically is entered in the session that is running on the console. The console can be switched to other terminals via the :CONSOLE command. To a certain extent, operator commands can be distributed to specific users in two ways. First, one operator command, :ALLOW , enables specific users the ability to execute specific operator commands. Second, a user may :ASSOCIATE a device class. An :ASSOCIATEd user can execute operator commands for any device belonging to that class.

Operator commands are typically concerned with the moment-by-moment adjustment of the system. Operators can control specific devices (:UP, :HEADON), the system spooler (:OPENQ, :ALTSPOOLFILE), the system's job/session mix (:LIMIT, and :ABORTJOB), and they can satisfy operator requests (:REPLY, and :RECALL).

"OP" Commands are "System Supervisor" commands. They allow a System Supervisor, who has a major responsibility for the day-to-day management of the operating system, to optimize the performance of a system. A user with "OP" capability can execute commands like :TUNE and :ALLOCATE and can log on with ;HIPRI; hence, this capability should not be given lightly. "OP" commands are not operator commands as the name might suggest, but are system supervisor commands.

A Console Command must be performed on the console. To execute a console command, a A^C must be entered on the console. The process PROGEN is the root process for the running operating system, and all console commands execute on PROGEN's stack. The A^C on the console causes PROGEN to print a "= " prompt on the console and to post a READ on the console for the command input. Once a console command has been executed, the console becomes available for normal terminal usage.

It is easy to confuse these command classes. In summary: "OP" commands allow a System Supervisor to optimize a machine's performance, Operator Commands typically control the devices on a system (e.g. spooling), and Console Commands are most often used for shutting down the system.

Two New MPE Capabilities

Two new capabilities are now offered with MPE: NA (Network Administrator), which allows the user to administer the data communications network globally; and NM (Node Manager), which allows the user to manage data communications subsystems at the user's local node.

Disc Caching Commands

By Al Kondoff

Many MPE users will have a significant improvement in system performance as a result of disc caching. Currently, disc caching is available with MPE V/P and MPE V/E on High Performance Series 39, 42, 48, and 68. This new feature speeds up disc file reads by bringing larger pieces of the disc files into memory when needed by programs, and by retaining these pieces in memory so that they are available for repeated references. These repeated references avoid the overhead of additional input requests from the disc. Writes to disc files will also have improved performance, since cached writes need only be moved into cached memory, rather than waiting for the physical disc updates to complete. This allows programs to continue running while the physical disc updates occur.

Disc caching is most useful in systems with excess main memory and processor capacity. Available unused memory is consumed by cached disc domains. Disc references to these domains are resolved with CPU MOVE instructions rather than physical disc accesses.

The operation of disc caching is transparent to the programs running in a cached system. Process accounting on a caching system differs from process accounting on an uncached system, because processes will be charged for CPU time associated with disc caching. In general, CPU utilization increases and disc utilization decreases in a cached system. The system and user processes in particular, spend much less wall time waiting for disc file input/output to complete.

To control disc caching, four new commands have been added to the operator command set. These commands enable or disable disc caching per device, and monitor and control disc caching. In addition to these new commands, the FSETMODE intrinsic has been extended to allow refined control over disc caching management on a file basis, as opposed to a system-wide basis. The new commands and new intrinsics are described below:

STARTCACHE

```
:STARTCACHE ldev
```

This command is used to enable caching on a single disc and requires System Supervisor (OP) or System Manager (SM) capabilities. Each disc will require a separate invocation of the command. The parameter *ldev* is the logical device number of the device to be cached. This device must be a disc, and the system must have the caching option installed. When the system is started, caching is disabled by default. Caching can be started by :STARTCACHE commands included in the System Operator's startup UDC.

STOPCACHE

```
:STOPCACHE ldev
```

This command is used to disable caching on a single disc and requires OP or SM capabilities. Each disc will require a separate invocation of the command. The *ldev* is the logical device number of the device to have caching disabled.

CACHECONTROL

```
              {SEQUENTIAL=sequentn}
:CACHECONTROL {RANDOM=randomn     }[,...]
                           {YES}
              {BLOCKONWRITE={NO } }
```

PARAMETERS

sequentn

Assigns the number of sectors read from disc of a sequential file, if the information is not found in cache. The disc read will stop at extent boundaries and will fetch at least the amount requested. The number of sectors must be greater than 1, but less than 96 (inclusive). Default is 96 sectors.

randomn

Specifies the number of sectors read from disc of a random access file of a cache read, if the information is not found in cache. The disc read will stop at extent boundaries, and will fetch at least the amount requested. The number of the sectors must be greater than 1 , but less than 96 (inclusive). Default is 16 sectors.

BLOCKONWRITE

Specifies whether or not to block the process until posting of cache buffers to disc completes. The BLOCKONWRITE must be equal to YES or NO . If BLOCKONWRITE=YES , enabling the Serial Write Queue or disabling BLOCKONWRITE for a file is overridden. Default is NO .

The :CACHECONTROL command is used to tune the performance of caching on a running system. This command requires SM or OP capabilities. The :CACHECONTROL requires that at least one keyword be provided. Any two or all three keywords can be used in a single command, but each keyword can only be used once. The order in which the keywords are specified is not significant. If a keyword is specified more than once, the system will give a warning and the last value specified will be used.

When an I/O read request is made to read data that is not currently in memory (i.e. a read "miss"), the requested portion of the disc must be fetched into memory. Disc caching will use information about the file extent size and access type (random or sequential) in its fetch strategy to maximize the "hit" ratio; that is, subsequent read requests may be satisfied by a disc domain fetched on a miss.

The size of the fetch will always be limited by the extent size of the file. Within this limitation, the disc domain fetched will be equal to the greater of the following two sizes:

- The requested size.

- The largest integer multiple of the requested size that is smaller than the value specified by the *sequentn* or *randomn* parameters.

One important factor in maximizing disc caching performance is achieving a high hit ratio on read requests, so changing the value of the *sequentn* or *randomn* caching parameters may help increase the hit ratio for user I/Os. A poor choice for either or both of these parameters, however, can decrease the hit ratio.

When disc caching is enabled on your system, all user I/Os will complete immediately after moving the data to or from a disc domain, eliminating the wait for a physical I/O to complete. In other words, I/Os will now complete at the processor speed, not the disc speed. When the I/O request is for a write, the physical write to post the disc will be initiated after the move to the disc domain is complete. However, the physical post will complete in the background while the process continues. Thus, write requests look completed to the user process before the disc has been updated. By specifying YES to the BLOCKWRITE parameter, all user write requests will not show complete until the domain has been posted to the disc. This guarantees that the information is on disc when the request completes; however, this does eliminate a performance benefit (i.e. completion at processor speed, not disc access speed) for writes that disc caching normally provides.

SHOWCACHE

```
:SHOWCACHE
```

The :SHOWCACHE command provides a brief description of caching performance. No capabilities are required. For example enter:

```
:SHOWCACHE

DISC   CACHE     READ WRITE         PROCESS                 % OF    CACHE
LDEV   REQUESTS  HIT%  HIT%  READ%  STOPS        K-BYTES  MEMORY  DOMAINS
-------------------------------------------------------------------------

1      105542    89    63    86     9897         69         1     51
2      191487    91    54    55     10794        714        17    225
3      99868     85    87    21     14895        577        14    63
4      39416     90    45    55     2371         117        2     25
```

```
Total 436313     90     67     55    37957        1477          37     364

Data overhead is 94K bytes, 49% of user I/Os eliminated.
Sequential fetch quantum is 96 sectors.
Random fetch quantum is 16 sectors.
BLOCKONWRITE=no
```

The following descriptions explain the above summary, since disc caching was enabled on the device.

COLUMN HEADING **MEANING**

DISC LDEV — Logical device number(s) of the disc(s) which is being cached. Read across the row to see the other entries on a per disc basis. The bottom row is a total across all discs.

CACHE REQUESTS — Tally of the number of logical I/O requests against caching. This is a double word counter, and will reset to zero automatically when caching is started on the disc or when the counter overflows.

READ HIT% — Cumulative percentage of logical disc I/O in the system for which the request was a read request, and the needed data was in a main memory cache domain when the read request occurred.

WRITE HIT% — Cumulative percentage of logical disc I/O in the system for which the request was a write request, and the needed data was in a main memory cache domain when the write request occurred.

READ% — Percentage of logical disc I/Os in the system which were read requests.

PROCESS STOPS — Number of times a process was stopped in order for a cache I/O request to complete. Process Stops occur due to read misses, or due to write hits in which the cached region is currently being posted to disc. This is a double word counter which will reset to zero, if it overflows.

K-BYTES — Count of the number of thousands of bytes (1024-byte groups) which are currently being used for main memory cache domains.

% OF MEMORY — Percentage of the total memory in the machine which is currently allocated for use by cache domains.

CACHE DOMAINS — Number of cache domains in memory.

Data overhead is the current dynamic data structure overhead used to support disc caching. Data overhead does not include the portions of disc, as reflected in the K-BYTES column, in main memory. Also note the current values for the three caching parameters, which can be altered via :CACHECONTROL.

FSETMODE INTRINSIC

The FSETMODE intrinsic has been expanded to include the BLOCKONWRITE and "Serial Write Queue" parameters. The following syntax illustrates the FSETMODE intrinsic as it specifically relates to disc caching. For additional information on the FSETMODE intrinsic, please refer to the MPE V Intrinsics Reference Manual (32033-90007).

SYNTAX

```
                IV        LV
FSETMODE (filenum,modeflags);
```

PARAMETERS

filenum *integer by value*
A word identifier supplying the file number of the file to which the call applies.

modeflags *logical by value*
A 16-bit value that denotes the access mode options in effect, as described below.

Bit (14:1) - BLOCKONWRITE for this file. This bit has the following settings:

=1 Enabled (YES).

=0 Disabled (NO). Default.

Bit (15:1) - Serial Write Queue. This bit has the following settings:

=1 Enabled for writes (YES).

=0 Disabled for writes (NO). Default.

The FSETMODE intrinsic has been extended to allow access to the MPE global Serial Write Queue and BLOCKONWRITE parameter on a file-by-file basis. By enabling the Serial Write Queue on a file, all write requests are guaranteed to be performed in the order they were issued. The Serial Write Queue does this to preserve integrity while allowing the performance benefits of uninterrupted process execution (BLOCKONWRITE=NO). Since this option may have an affect on total system performance, please consult your SE or SEO (Service Engineering Organization) before enabling it. The Serial Write Queue is globally disabled if the :CACHECONTROL BLOCKONWRITE=YES was specified, since this causes all processes on the system to wait for their physical disc I/O to complete.

BLOCKONWRITE management can also be controlled on a file-by-file basis with the FSETMODE intrinsic. If BLOCKONWRITE is locally enabled, the user's application will not be notified of a disc write completion until the physical write operation has completed. Therefore, the BLOCKONWRITE option guarantees "commitment" of the transaction to disc prior to completion notification to the user's application. The FSETMODE setting of an individual file will be overridden if the :CACHECONTROL BLOCKONWRITE=YES is specified.

A combination of these two options in FSETMODE provides complete integrity and transaction commitment notification with disc caching, with minimal performance impact. Many disc writes can be performed with BLOCKONWRITE=NO , Serial Write Queue enabled, and guarantee integrity. On the final disc write of a transaction, both localized BLOCKONWRITE=YES and Serial Write Queue enabled can be specified to provide both integrity and total transaction commitment notification.

ADDITIONAL CONSIDERATIONS

With disc caching enabled, the system can have many writes pending against the discs, as opposed to several writes in noncached MPE. In the event of a system failure, the potential for losing writes to disc is increased. Again, it is possible to globally enable :CACHECONTROL BLOCKONWRITE=YES to limit the write exposure, so it is equivalent to noncached MPE, but at the expense of some caching performance.

MPE-managed subsystems, such as IMAGE and KSAM, have been enhanced to utilize disc caching synchronization and transaction commitment notification. If other user-managed disc files are sensitive to the order in which writes are performed, two options are available. Application programs can be modified to call FSETMODE for these sensitive files, or the global CACHECONTROL BLOCKONWRITE=YES can be specified after caching is enabled. The :CACHECONTROL option will negate a portion of the benefit provided by disc caching, but will provide total compatibility with noncached MPE.

:RESTORE Command Enhanced

Edited By Mike Clark

The MPE :RESTORE command has been enhanced on the G.00.00 release of MPE V/E in an ongoing effort to better meet our customers' reliability and functionality requirements. The :RESTORE command now provides recovery from tape errors. File set handling capabilities have been extended, and the functionality has been increased. This article details the enhancements to :RESTORE . The following is an excerpt from the first edition of the MPE V Commands Reference Manual (32033-90006). Note that only the new parameters are defined in any detail here.

The :RESTORE command syntax is:

```
                         {fileset[-fileset][,fileset[-fileset]][,...]}
:RESTORE [restorefile][;{!indirectfile                              }]

   [;DEV=device]

   [;SHOW=[showparm[,showparm[,...]]]

   [;FILES=maxfiles]

     {LOCAL                         }
   [;  {GROUP=groupname    }         ]
     {                      }[;...]}
       {ACC[oun]T=acctname }

               {GROUP  }
    [;CREATE[={ACCT    }[;...]]
               {CREATOR}

    [;CREATOR=[username]]

      {KEEP  }
    [;{NOKEEP}]

      {OLDDATE}
    [;{NEWDATE}]

                {QUIT}
    [;ONERR[or]={SKIP}]
```

Default Tape Name

In previous releases of MPE it was necessary to specify the name of a back-referenced tape or serial disc file in the :RESTORE command. Now if the user does not specify a *restorefile* , then a default tape name will be used. The default tape name is equal to the user's logon identification and the device is TAPE . For example if the user is logged onto TOM.MGR then the tape request will be for " TOM ". No file equation may be used. Here are a few examples of using the default tape name:

```
:RESTORE              ** Restores all files. **
:RESTORE;;SHOW        ** Restores all files with SHOW option. **
:RESTORE;@.PUB.SYS    ** Restores @.PUB.SYS. **
```

Expanded SHOW Option

The information reported by ;SHOW has been expanded by the addition of the following options:

SHORT

For each file restored using the :RESTORE command, the following is printed on the user's terminal (default). The options include: file name, group name, account name, logical device number, reel numbers, file size (in sectors) and file code. The default is SHORT if $STDLIST is less than 99 characters.

LONG

Displays same information as SHORT , plus: record size, file type, EOF , file record limit, blocking factor, extents allocated and maximum extents of files restored. If the list device for SHOW has a record size greater than 100 characters, this option is selected by default.

DATES

Causes SHOW to display the creation date, last access date, and last modification date of restored files.

SECURITY

Causes SHOW to display the creator and file access matrix of all files restored.

OFFLINE

Causes SHOW output to be sent to the system line printer in addition to being displayed on the user's terminal. Note that output can be redirected to another output device by using a :FILE command. (Formal designator is OFFLINE .)

Although you cannot specify both LONG and SHORT, you may use more than one *showparm* if they are separated by commas.

Tape Error Recovery

In previous versions of :RESTORE , any error encountered while reading from tape would cause the :RESTORE operation to abort. Thus, if :RESTORE encountered a tape parity error on the first file on the tape it may have been impossible to get other files restored from the tape.

With the enhanced version, you can specify whether or not error recovery is desired by the use of the ONERR keyword:

```
           {QUIT}
;ONERR[or]=
           {SKIP}
```

If ONERR is not specified, the defaults are SKIP for unlabeled tapes and QUIT for labeled tapes. If the QUIT option is selected, the :RESTORE operation aborts upon sensing a tape error. The SKIP option causes :RESTORE to do a file-skip-forward past the tape error, resynchronize, and begin reading again from the tape. One or more files may be lost during resynchronization. The ONERR=SKIP cannot be specified for labeled tapes.

Improved Disc/Directory Error Tolerance

If a disc or directory error is encountered while updating the directory, updating the Disc Free Space tables, or writing the data to the disc file, the error is reported to SYSLIST (defaults to $STDLIST) and :RESTORE will continue. If a previous version of the file existed, it will still be on the system. Because of this recovery ability, the user must have enough space on disc to restore a file without purging the old version (if this is a problem the user can purge the file using the :PURGE command).

Messages

The message-handling facility of :RESTORE has been enhanced to print messages as they are generated, rather than displaying a list of messages at the end of the :RESTORE operation. Interactive users can now monitor the progress of :RESTORE and see error messages as they occur. Thus, you can abort :RESTORE if, for example, you discover that an important file is not on the tape at the time of the restore. If the ;SHOW option was specified, a message is displayed as each file is restored.

Improved Error Messages

Previously, when :RESTORE encountered a problem of any kind, it aborted with the error message:

```
ERROR ACCESSING TAPE - RESTORE STOPPED (CIERR 1026)
```

Since this message could be generated by several different causes, diagnosis of and recovery from the problem was often difficult. The enhanced version of :RESTORE generates an error message appropriate to the particular error encountered, making it easier to determine the correct recovery procedure.

Indirect Fileset Specification

The :RESTORE command now allows the use of indirect files to specify file sets and options. This feature is useful, for example, in the case where it is necessary to restore the same files over and over, or when the Command Interpreter limitation of 268-character commands is a problem. When using an indirect file, each line can contain both file sets and options, but only options can appear after the first semicolon (;) on each line. (Indirect file set specification was introduced on :STORE with C/D.01.00.) For example, suppose you have a file INDREST containing:

```
@.TOM.MGR, @.PUB.SYS
@.@.support; SHOW=LONG,DATES
```

To use INDREST as the list of files to restore, enter:

```
:RESTORE*TAPE;!INDREST
```

The -fileset Parameter

The *filesetname* parameter has been expanded to allow for elimination of a particular file set or subfile set with the use of the new *-fileset* parameter. You cannot nest the *-fileset* parameter. There can only be one *-fileset* for each fileset. A negative sign (-) must precede this parameter. For example:

```
    :RESTORE *TAPE;@.@.@ - @.PUB.SYS  ** Restores all files except those
                                          in the PUB group of the SYS account **
```

The *-fileset* parameter was introduced on :STORE with C/D.01.00.

The LOCAL Option

In previous versions of :RESTORE it was often impossible to restore files from a different system because of different directory structures. In many cases the desirable solution would be to restore the file into the user's logon group and account. The :RESTORE command now provides this function with the LOCAL option. If the LOCAL option is used, all the files that are restored will be placed in the user's logon group and account and the creator identification of the file will be changed to the user's logon identification. In order to be able to restore a file with the LOCAL option, the user must have read access to the file on the tape. The LOCAL option to the :RESTORE command replaces the program RESTORE.PUB.SYS .

The GROUP and ACCOUNT Options

These two options provide a more general ability to move files into different groups and accounts than the LOCAL option. If the GROUP option is specified (need System Manager or Account Manager capabilities), then all files that are restored will have their groups changed to the *groupname* provided. If the ACCOUNT option is specified (need System Manager capability; ACCOUNT may be specified), then all files that are restored will have their accounts changed to the *accountname* provided. Both the GROUP and the ACCOUNT options may be placed on the same :RESTORE command; however, if the LOCAL option is used then neither the GROUP nor the ACCOUNT options may be used.

The CREATOR Option

In previous versions of :RESTORE , a file could not be restored if the creator of that file was not in the directory. This was often frustrating because the user did not know who created the file. In an effort to respond to customer suggestions, :RESTORE has been changed so that files will be restored even if the creator does not exist. If the user does not specify the CREATOR option then the creator of the restored file is determined as follows:

1. The creator identification from the file label as it appears on tape if that creator exists.

2. The user identification of the session performing the :RESTORE if the file is in the same account that the user is logged on to.

3. The string " RESTORE ". A blank must follow :RESTORE.

If the user does specify CREATOR but does not specify a *username* with it, then the creator identification from the file label will be used. The creator must exist on the system or the file is not restored. If the user specifies CREATOR and provides a *username*, then all files will have their creator identifications changed to that *username*. If that *username* does not exist, then the file is not restored. The user may create any missing creators by using the CREATE option described in the next section.

The CREATE Option

A new option, CREATE , has been added to :RESTORE that allows users to restore files whose group, account, or creator does not exist in the directory. When this option is specified, :RESTORE creates the needed directory entries. If an entry is created, a message is sent to SYSLIST (default $STDLIST). The user can also specify what type of directory entries can be built. In order to create accounts, the user needs System Manager (SM) capability. In order to create groups or creators, the user needs Account Manager (AM) capability. If the user specifies CREATE , but does not specify any create options, then :RESTORE will create directory entries based on the user's capabilities. Accounts are created with Account Manager "MGR", and all other options are set to the default values. Groups and users are created with all options set to the default values. Examples of the CREATE option are:

```
:RESTORE *T;;CREATE=CREATOR    ** Creates only missing creators. **

:RESTORE *T;;CREATE      ** Creates directory entries based on capabilities. **

:RESTORE *T;;CREATE=GROUP,ACCT,CREATOR  ** Creates any directory entry
                                           missing.  Need AM and SM. **
```

The KEEP and NOKEEP Options

If, while doing a :RESTORE an identically named file exists in the system, then :RESTORE can take one of two actions. If the user specifies the KEEP option then the file on disc is kept and the corresponding tape or serial disc file is not restored. If the user specifies the NOKEEP option then the file on disc is purged and the corresponding tape or serial disc file is restored. If the file cannot be successfully restored, :RESTORE recovers the old disc file. Default is NOKEEP .

The OLDDATE and NEWDATE Options

The file system keeps four dates in the file label of each file. These dates are the creation date, modification date, last access date, and the allocation date. The allocation time is also kept in the file label. If the user specifies the NEWDATE option then all the dates and times are changed to the date and time that the :RESTORE takes place. If the user specifies the OLDDATE option then the modification date and the last access date are set to the dates that were in the file label when the file was stored.

Implementation

The enhanced version of :RESTORE now resides in a program file (STORE.PUB.SYS), rather than a set of procedures with no global memory. It is invoked by both the :STORE and :RESTORE commands. It is advisable to use :ALLOCATE STORE.PUB.SYS in order to speed up the loading process and reduce the use of system resources. If you are restoring from a large tape or disc, such as a :SYSDUMP tape, you must include the *maxfiles* parameter or the file(s) will not be restored. The value of *maxfiles* should be at least as large as the maximum number of files on the device from which you are restoring.

As a result of the new structure, :RESTORE can now be invoked via the :RUN command. You can use the INFO= parameter to specify file sets and options; the :RESTORE command will execute with no further interaction. An example follows:

```
:RUN STORE.PUB.SYS;INFO="RESTORE *TAPE;@.@.@;SHOW "
```

If no INFO= parameter is specified, the <-- prompt will appear. Acceptable responses are: a complete :STORE command, a complete :RESTORE command, an MPE command preceded by a colon (:), or :EXIT.

Notes on JCWs

When invoking :RESTORE via the Command Interpreter (e.g., :RESTORE *T;@) the system defined JCW CIERROR will be set to the value 1091 if the :RESTORE command aborts for any reason. You can check this value after the :RESTORE aborts by entering the command :SHOWJCW.

When invoking :RESTORE via the :RUN command, the system defined JCW CIERROR will not be changed even if the :RESTORE aborts.

Enhancements to :STORE

Edited By Mike Clark

Two enhancements have been made to the :STORE command. A PURGE option has been added to allow the user to purge files automatically after they have been stored. The syntax for :STORE has also been improved to allow for a default tape name, if the user does not supply one.

PURGE Option

The PURGE option is a new feature of :STORE which purges all files that were stored. If a file is not stored successfully, the file will not be purged. Files are purged after the entire :STORE operation. Therefore, if a catastrophic error occurs (such as system failure or power failure), no files will be purged. Here are a few examples of the :STORE command using the PURGE option:

```
STORE @.RYALL.KSE;*TAPE;SHOW;PURGE       ** Stores all files in a group and
                                            then purges them. **

STORE @.@.@;*TAPE;DATE<=12/31/82;PURGE  ** Stores all files not accessed this
                                           year and then purges them. **
```

If a file has a negative file code, then :STORE will not be able to purge it (even if a user has Privileged Mode). If a file has a lockword, the user will be prompted for the lockword (the :STORE command will hang for each prompt unless the user has System Manager capability). If a file could not be purged, a File System Error message is sent to the user, which will state that the file was not purged.

Default Tape Name

In previous releases of MPE, it was necessary to specify the name of a back-referenced tape or serial disc file in the :STORE command. Now it is not necessary to provide this file name. If the user does not specify the file name, :STORE will provide a default tape name. Here are a few examples of using the default tape name:

```
:STORE                   ** Store all files in logon group. **

:STORE A,B,C             ** Stores files A, B, and C. **

:STORE @.PUB.SYS;;SHOW   ** Stores files in PUB.SYS using SHOW. **
```

The default tape name is equal to the user's logon identification and the device is TAPE . For example, if the user is logged onto TOM.MGR , then the tape request will be for " TOM ". No file equation may be used.

New :OPENQ and :SHUTQ Commands

By Elizabeth Wei

Two new operator commands, :OPENQ and :SHUTQ , now allow explicit control of spooling queues, or the ability to create spoolfiles when a device is referenced by a process. These commands already exist as options to other spooler commands, as in :STARTSPOOL and :STOPSPOOL . This implementation will remain unchanged. The new individual commands were added for these reasons: first, to enable control of spool queues without affecting the operation of a spooler process, and second, to allow access to spool queues for which no spooler or real physical device exists.

The syntax for these commands is:

```
:OPENQ {ldev     }
       {devclass}
```

```
:SHUTQ {ldev     }
       {devclass}
```

The :OPENQ command opens the spool queue for the specified logical device (*ldev*) or device class (*devclass*). Similarly, the :SHUTQ command closes the spool queue for the named logical device or device class. The *ldev* must be an output spoolable device, and *devclass* must contain at least one output spoolable device. The device must be configured on the system, although it need not physically exist. A spooler process for the device is not required.

The commands :OPENQ and :SHUTQ , are operator commands. A user wishing to execute them must be the master operator, be :ALLOWed the use of this command, or be :ASSOCIATEd to the device or class specified. Both commands can be executed programmatically, from a session or a job, or in break. The commands are not breakable. For example:

To open the spool queue for logical device 6, enter: :OPENQ 6

To shut the spool queue for device class LP, enter: :SHUTQ LP

As an example of typical use, suppose spoolfiles are being created faster than a certain device is able to print them. The :SHUTQ with the *ldev* parameter would be used to close the spool queue for that device. When the printer has finally caught up with its backlog and can again be made available to users as a spooled device, :OPENQ with the *ldev* parameter would be executed. If the queue is shut after a spoolfile has been created, the spoolfile will finish printing.

Possible errors and warnings returned by these commands are:

```
OPENQ HAS EXACTLY ONE PARAMETER, A DEVICE NUMBER OR CLASS. (CIERR 3262)
SHUTQ HAS EXACTLY ONE PARAMETER, A DEVICE NUMBER OR CLASS. (CIERR 3263)
EXPECTED LDEV OR DEVICE CLASS. (CIERR 3264)
DEVICE QUEUE ALREADY OPEN. (CIWARN 3265)
DEVICE CLASS QUEUE ALREADY OPEN. (CIWARN 3266)
DEVICE QUEUE ALREADY SHUT. (CIWARN 3267)
DEVICE CLASS QUEUE ALREADY SHUT. (CIWARN 3268)
```

Privileged Mode DEBUG Changes in MPE V/E

By Jessy J. Hsu

The syntax of privileged DEBUG commands has been changed for MPE V/E. The new syntax applies to those systems having new microcode and running MPE V/E. The syntax of non-privileged DEBUG commands remain unchanged.

Several of the commands in DEBUG specify the segment number. In most cases, this segment number, specified by the user, is a logical segment number within a Segmented Library (SL) or program file, and thus the number does not change if the segment mapping scheme changes. However, the BA , CA , RA , DCO , FCO , and UCO commands (these are privileged commands) specify an absolute segment number. Since the absolute program segment number starts at 1, MPE V/E users have to tell DEBUG whether absolute segment number is a Code Segment Table (CST) entry or a Code Segment Table Expansion (CSTX) entry. The X parameter has been added to allow the user to specify an absolute CSTX segment number. In MPE IV and MPE V/P (which do not have the new microcode), absolute program segment numbers start at %301.

The absolute CSTX number changes to DEBUG syntax are as follows:

Privileged B Command

```
         [A  [X]]
         [G     ]                          [cond ]
B [[pin.][P     ] segment.] offset [:[@]   [count]] [,...]
         [S     ]
```

Privileged C Command

```
         [A  [X]]
         [G     ]
C [[pin.][P     ] segment.] offset [,...]
         [S     ]
```

Privileged F Command

```
  {CO [X]}
F {DA    } segment
```

Privileged DCO Command

```
DCO [X] segment [+offset] [,count][,MODE]
```

Privileged R Command

```
           [A  [X]]
           [G     ]                      [cond  ]
R [[pin.]          segment.] offset [:[@]        ] [,...]
           [P     ]                      [count ]
           [S     ]
```

Privileged U Command

```
  {CO  [X]}
U           segment
  {DA      }
```

Two additional display registers have been added to the display of the DR command: MF and MDST. The register MF has been added to the T command. MF=1 means the segment is physically mapped, MF=0 logically mapped. MDST is the mapping DST number if "0" means process does not have a map.

The following examples show the use of these changed DEBUG commands. MPE IV and MPE V/P DEBUG commands are also listed for comparison.

Task	MPE V/E with new microcode	MPE IV & MPE V/P with old microcode
Set a breakpoint at location 33 of CST #10	BA 10.33	BA 10.33
Set a breakpoint at location 33 of CSTX #10	BA X10.33	BA 310.33
Clear a breakpoint at location 33 of CST #10	CA 10.33	CA 10.33
Clear a breakpoint at location 33 of CSTX #10	CA X10.33	CA 310.33
Resume execution and set a breakpoint in CST #10	RA 10.33	RA 10.33
Resume execution and set a breakpoint in CSTX #10	RA X10.33	RA 310.33
Freeze CST #10	FCO 10	FCO 10
Unfreeze CST #10	UCO 10	UCO 10
Freeze CSTX #10	FCO X10	FCO 310
Unfreeze CSTX #10	UCO X10	UCO 310
Display CST #10 word 20	DCO 10+20	DCO 10+20
Display CSTX #10 word 20	DCO X10+20	DCO 310+20

Three New Commands Support COBOL II

:COBOLII, :COBOLIIPREP, :COBOLIIGO

By Geoff Ralston

Three new MPE commands facilitate the use of the COBOL '74 compiler "COBOLII". (The COBOL '68 compiler "COBOL" is scheduled for obsolescence in May 1987.)

The `:COBOLII` command compiles a program using the COBOL '74 compiler. The `:COBOLIIPREP` command compiles and prepares a USL file. `:COBOLIIGO` compiles, prepares, and runs the COBOL program.

The *WKSP* parameter is the designator of an HPToolset workspace. COBOLII utilizes the formal file designator `COBWKSP`. For further information, refer to the MPE V Commands Reference Manual (32033-90006).

The syntax for these three new commands follows:

```
:COBOLII [textfile][,[uslfile][,[listfile][,[masterfile]
           [,newfile]]]][;INFO=quotedstring][;WKSP=workspacename]
```

```
:COBOLIIPREP [textfile][,[progfile][,[listfile][,[masterfile]
               [,newfile]]]][;INFO=quotedstring][;WKSP=workspacename]
```

```
:COBOLIIGO  [textfile][,[listfile][,[masterfile][,newfile]]]
              [;INFO=quotedstring][;WKSP=workspacename]
```

New Devices Can Be Used as Private Volumes

By Jan Helmbolt

The HP 7911, 7912, 7914, and 7935 disc drives can now be specified as Private Volumes with the :NEWVSET and :ALTVSET commands. The *type* parameter for both commands can now be HP7911 , HP7912 , HP7914 , and HP7935 , in addition to other names. (Refer to the MPE V System Operation and Resource Management Reference Manual (32033-90005) for a syntax listing of these commands.)

The :LISTVS command now lists the volume set(s) specified and the members associated with a given volume set. The :VINIT and :DSTAT commands now support the new disc devices.

These enhancements are described in the MPE V System Operation and Resource Management Reference Manual (32033-90005).

UDC Initialization Error Affects Only Its Level

By Geoff Ralston

Every time a user logs on, the system initializes the user's User Defined Command (UDC) directory. UDCs can be established to apply to one user only, the whole account, or the whole system.

Occasionally, an error may be detected when the UDC directory is being initialized at logon. This error could be caused by an invalid UDC-header format in the UDC file (i.e., an illegal UDC name). The error could also be caused by MPE being unable to open the UDC file, perhaps due to its having been purged. Until now, a UDC error at any level prevented the user from accessing any user, account, or system UDCs.

Now, only the UDC level in which an error occurs will fail to be initialized. If a user-level UDC error occurs at logon, the user will still have access to account-level and system-level UDCs. If an account-level UDC error should occur at logon, the user will still have access to user-and system-level UDCs.

For example, `JOHN.DOE` has used `:SETCATALOG` on a valid UDC file. While he is logged off, `MARY.DOE` purges his UDC file. When John logs on again, his user-level UDCs will not work, but he still has access to account- and system-level UDCs.

For complete details on the definition and use of User Defined Commands, refer to the MPE V Commands Reference Manual (32033-90006).

:CONSOLE Command Displays Additional Message

By Geoff Ralston

Formerly, when the console was assigned to a new logical device, only the old console displayed an informational message. Now, both the old and new console screens display the message:

```
CONSOLE HAS BEEN SWITCHED FROM LDEV old TO LDEV new
```

GETDSEG Intrinsic Changes

By Greg Grimm

Because of the increased number of possible segment entries in the Data Segment Table (DST) for MPE V/E, the GETDSEG intrinsic was modified. GETDSEG has the following format:

```
GETDSEG (index , length , id );
```

The *index* parameter is used to identify which data segment was assigned by MPE for the user. When GETDSEG is called in User Mode, all subsequent calls to intrinsics that use *index* must now be in User Mode. Likewise, when GETDSEG is called in Privileged Mode, all subsequent calls to intrinsics that use *index* must now be in Privileged Mode.

The following conditions caused this change to be necessary:

- When GETDSEG is called in User Mode, *index* is a logical index of the assigned data segment. If an error is found, *index* will be set to %2000-%2003.

- When GETDSEG is called in Privileged Mode, *index* is the actual segment entry number for the data segment that was assigned.

Suppose a User Mode process calls GETDSEG and an error of %2001 is assigned to *index*. The process then calls GETPRIVMODE to acquire Privileged Mode. Now, if the process uses *index* as a privileged user, it could be interpreted as a valid data segment entry number for MPE V/E, not the error returned to the User Mode process that invoked GETDSEG. Thus, MPE will check that the process will use *index* as described above.

New Intrinsic JOBINFO

By Larry Cargnoni

The JOBINFO intrinsic is new for MPE V/E, and will execute only on an HP 3000 supporting MPE V/E. It enables a standard user to access session- and job-related information. Previously, most of this information was accessible only through MPE commands and the WHO intrinsic. The JOBINFO intrinsic is patterned after the FFILEINFO and CREATEPROCESS intrinsics. It is callable from any language, and may be used in software that performs security checks, job stream polling, system accounting, and job/session communication. The JOBINFO intrinsic provides access to information related to any job/session that is current to the system. This intrinsic is expandable, and is written so that the addition of further functionality will be straightforward.

SYNTAX

```
            IV    D        LA           IV        LA          I       OV
JOBINFO(jsind,JS#nnn,status  [,itemnum1,item1,errornum1]
                             [,itemnum2,item2,errornum2]
                             [,itemnum3,item3,errornum3]
                             [,itemnum4,item4,errornum4]
                             [,itemnum5,item5,errornum5]);
```

PARAMETERS

jsind

integer by value (required)
An integer indicating whether the *JS#nnn* is a session or job:

1= *JS#nnn* is a session.

2= *JS#nnn* is a job.

JS#nnn

double (required)
A double value, 32 bits, identifying the job or session for which information will be retrieved.

status

logical array (required)
A two word logical array reporting the overall success/failure of the call. Only the first word contains significant information.

0= Successful call. All *errornums* equal zero.

1= Semi-successful call. One or more *errornums* were returned with nonzero values.

2= Unsuccessful call. All *errornums* were returned with nonzero values.

3= Unsuccessful call. Syntax error in calling sequence.

4= Unsuccessful call. Unable to retrieve *JS#nnn/S#nnn* .

5= Process died during the start of retrieval.

itemnum

integer by value (optional)
Cardinal number of the item desired. This specifies which item value is to be returned (refer to "ITEM#" in Table 1).

item

logical array (optional)
Name of a reference parameter (whose data type corresponds to the data type for the desired information) to which the desired information is returned (refer to "ITEM" in Table 1).

errornum

integer (optional)
A returned integer specifying the success or failure of the retrieval of each item. The returned values are:

0= Successful information retrieval.

1= Invalid *itemnum* (item number).

2= Desired information not pertinent to the given *JS#nnn* (eg., user specifies a session number and wishes to know if a job had RESTART option).

3= User has insufficient capability to access this information.

4= The desired information is no longer available (eg., when spoolfiles disappear).

SPECIAL CONSIDERATIONS

A user without System Manager (SM) or Account Manager (AM) capability can only retrieve information about the jobs/sessions logged on under the user name and account. A user with AM capability but not SM capability will be restricted to information concerning his account sessions and jobs; a user with SM capability will be able to retrieve information concerning all sessions and jobs. The exception to the above security is access to items which are normally available to a user, through MPE commands, who does not have any special capabilities.

CONDITION CODES

There are no condition codes in the traditional sense, but the *status* parameter can be thought of as a condition code.

The *status* parameter returns a number representing the overall status of the call. The *errornum* parameter returns the status of the individual accesses *items* and *itemnums* . Combinations of successful and unsuccessful data retrievals could be returned from the same call. For example, a user who does not have System Manager or Account Manager capabilities writes a program with JOBINFO . The JOBINFO intrinsic retrieves the jobfence and the current job step of access user. Upon the return of JOBINFO , the parameter *status* will return a 2 (semi-successful call). The call is not successful since the *errornum* corresponding to the jobfence access will be 0 (successful retrieval) and the *errornum* corresponding to the current job step access will be a 3 (insufficient capability).

Table 1. Item Descriptions

ITEM#	ITEM (information returned)	DATA TYPE
1	[JSNAME,]user.account (See note 1)	LA
2	session/job name (See note 2)	LA
3	user name (See note 2)	LA
4	user logon group (See note 2)	LA
5	user account (See note 2)	LA
6	user home group (See note 2)	LA
7	session/job introduction time (See note 3)	LA
8	session/job introduction date (See note 4)	LA
9	input ldev/class name (See note 2)	LA
10	output ldev/class name (See note 2)	LA
11	current job step (See note 5)	LA
12	current number of jobs	I
13	current number of sessions	I
14	job input priority	I
15	job/session number	D
16	jobfence	I
17	job output priority	I
18	number of copies	I
19	job limit (system)	I
20	session limit (system)	I
21	job deferred (See note 6)	I
22	main PIN - CI PIN for job/session	L
23	original job-spooled (See note 6)	L
24	RESTART option (See note 6)	L
25	sequenced - job (See note 6)	L
26	term code (See note 7)	L
27	CPU limit	L
28	session/job state (See note 8)	L
29	user's local attributes	D
30	$STDIN spoolfile number (See notes 9 & 10)	I
31	$STDIN spoolfile status (See notes 9 & 11)	I
32	$STDLIST spoolfile number (See notes 9 & 10)	I
33	$STDLIST spoolfile status (See notes 9 & 11)	I
34	length of current job step of item number 11	I
35	:SET $STDLIST=DELETE invoked (See note 12)	L
36	Job Information Table data segment number	L

Table 1 (continued). Item Number Notes

1. Can be used as an input or output parameter. If it is used as an input parameter, a maximum of 26 characters plus one for a Binary 0 terminator is allowed. The input string must be in the form of [JSNAME,]USER.ACCT. The wild card character (@) is not allowed. If it is used as an output parameter, the logical array must be 13 words long. Output is left-justified and padded with blanks.

2. An output parameter. Logical arrays must be 4 words long. Output is left-justified and padded with blanks.

3. Returns a 32-bit double word in a form to be used by the FMTCLOCK intrinsic.

4. Returns a 16-bit logical word in a form to be used by the FMTCALENDAR intrinsic.

5. Returns a maximum of 283 characters. It is the image of the command currently executing. The logical array must be long enough to accommodate the expected command image.

6. Returns the values: 0 = No.
1 = Yes.

7. Returns the values: 0 = Regular terminal.
1 = Regular terminal with special log on.
2 = APL terminal.
3 = APL terminal.

8. Returns the values: 2 = Executing.
4 = Suspending.
32 = Wait.
48 = Initialization.

9. Returns data for current jobs and sessions. $STDIN/$STDLIST files only.

10. Returns the spoolfile number as an integer.

11. Returns the values: 0 = Active.
1 = Ready.
2 = Open.
3 = Locked.

12. Returns the values: 0 = $STDLIST will be saved.
1 = :SET $STDLIST=DELETE is invoked.

All *itemnum* and *item* parameters are output parameters with one exception. Item number 1 can be used for an input and output parameter. Item number 1 is an input parameter only if the user is identifying a job or session by parsing a character string. The format of the character string is:

```
[JSNAME,]USER.ACCT
```

(Refer to the MPE V Intrinsics Reference Manual (32033-90007).) Otherwise, it is an output parameter. The number of characters returned is twenty-six or less. The returned string will be left-justified and padded with blanks:

```
[JSNAME,]USER.ACCOUNT
```

A useful program would be to poll a user's :STREAM job to determine what job step it is currently executing. The following piece of pseudo code could be used to accomplish this:

```
$CONTROL USLINIT
BEGIN
EQUATE JOB = 2;
DOUBLE JOB'NUMBER;
INTEGER CUR'JOB'STEP'LEN := 0,
        LEN              := 0,
        I                := 0,
        LENUA            := 0,
        ERROR1           := 0,
        ERROR2           := 0,
        ERROR3           := 0,

LOGICAL ARRAY
        STATUS(0:1),
        JOB'NAME(0:3);

LOGICAL ARRAY RESULT(0:4) := "JOB STEP =";
LOGICAL ARRAY PROMPT(0:7) := "ENTER JOB NAME";

BYTE ARRAY JSNAME(0:26),
           CUR'JOB'STEP(0:79),
           MYACCT(0:7),
           MYUSER(0:7),
           BJOB'NAME(*) = JOB'NAME;

LOGICAL ARRAY LJSNAME(*) = JSNAME,
              LCUR'JOB'STEP(*) = CUR'JOB'STEP;

INTRINSIC JOBINFO, WHO, PRINT, READ;

SUBROUTINE COUNT( STRING, NUM'CHARS );
  BYTE ARRAY STRING;
  INTEGER NUM'CHARS;

  COMMENT:  COUNT returns the actual number of non-blank
            characters in STRING througH NUM'CHARS.  NUM'CHARS
            can not be greater that 8
            since STRING will be an account or a user name;
```

```
 BEGIN << Subroutine COUNT >>
  NUM'CHARS := 0;
  I := 0;
  WHILE I < 8 DO
   BEGIN
   IF STRING( I ) <> " "
    THEN NUM'CHARS := NUM'CHARS + 1;
   I := I + 1;
   END;
 END;  << Subroutine COUNT >>

<< Main procedure >>
MOVE JSNAME := 26(" ");
<<                >>
<< Prompt user for job name and read it >>
PRINT( PROMPT, 7, %40 );
IF <> THEN ...

LEN := READ( JOB'NAME, -8 );
IF <> THEN ...

<< Now must set up JSNAME in the format of [jobname,]user.acct >>
IF LEN <> 0
  THEN BEGIN
    MOVE JSNAME( 0 ) := BJOB'NAME,( LEN );
    MOVE JSNAME( LEN ) := ",";
    LEN := LEN +1;
    END;

WHO(,,,MYUSER,,MYACCT );  << Obtain user name and account >>
COUNT( MYUSER, LENUA );
MOVE JSNAME( LEN ) := MYUSER(0), ( LENUA );
LEN := LEN + LENUA;

JSNAME( LEN ) := ".";

LEN := LEN+1;
COUNT( MYACCT, LENUA );
MOVE JSNAME( LEN ) := MYACCT(0), ( LENUA );
<<Append a Binary 0 terminator>>
MOVE JSNAME (LENUA + LEN) := 0;

STATUS(0) := 0;
STATUS(1) := 0;
JOB'NUMBER := 0D;

<< Now call JOBINFO,  JSNAME will qualify the job to be looked at >>
JOBINFO( JOB, JOB'NUMBER, STATUS, 1, LJSNAME, ERROR1,
                                 11, LCUR'JOB'STEP, ERROR2,
                                 34, CUR'JOB'STEP'LEN, ERROR3);
```

```
IF STATUS( 0 ) <> 0
   THEN ERR'ROUTINE.......
   ELSE BEGIN
        << Tell the user what job step is currently being executed >>

        PRINT( RESULT, 5, %40 );
        IF <> THEN ...

        CUR'JOB'STEP'LEN := -CUR'JOB'STEP'LEN;

        PRINT( LCUR'JOB'STEP, CUR'JOB'STEP'LEN, %40 );
        IF <> THEN ...

        END;

END. << Program >>
```

For additional discussion refer to the MPE V Intrinsics Reference Manual (32033-90007).

FFILEINFO: 16 Bit LDEVs And Other Enhancements

By Gaetano J. Randazzo

Before the G.00.00 release of MPE V/E, the number of devices that could be configured on a system was limited to 255, a number that could be stored in 8 bits. This was reflected in the logical device number (*ldev*) parameter of FGETINFO and FFILEINFO (item number 6). These items returned the virtual logical device number in the upper 8 bits and the real device number in the lower 8 bits. One use of these items is to determine if the file was opened over a DS line. Another use is to reopen a file by logical device number that was originally opened by class. The logical device number obtained from these intrinsics is used to reopen the device (e.g., a tape).

The G.00.00 release of MPE V/E allows more than 255 devices to be configured. Since the logical device number may not fit into 8 bits, two new FFILEINFO items have been introduced. Item number 50 will return the real device number of the file and item 51 will return the virtual device number. Both of these items are full 16 bit integers. The old FGETINFO or FFILEINFO (item 6) used with the *ldev* parameter will be returned as follows:

- If both the real and the virtual logical device number (if it exists) are less than 255, then the item will be returned as before.
- If either item is greater than 255, then the entire return value will be zero. This means that as long as no devices are configured with logical device numbers greater than 255, then all existing applications that use these intrinsic items will work.

All programs that use these items should be altered to use FFILEINFO items 50 and 51 if there is the possibility that a logical device number, especially a virtual terminal, could be greater than 255.

The following items have also been added to the FFILEINFO Intrinsic:

Item #	Type	Description
52	Double	Last modification time (CLOCK format).
53	Logical	Last modification date (CALENDAR format).
54	Logical	File creation date (CALENDAR format).
55	Logical	Last access date (CALENDAR format).
56	Integer	Number of data blocks in a variable length file.
57	Integer	Number of the user label written to the file.
58	Integer	Number of opens for output.
59	Integer	Number of opens for input.
60	Integer	Terminal type, defined as:

0 - File's associated device is not a terminal.
1 - Standard hardwire or multi-point terminal.
2 - The terminal is connected via a phone-modem.
3 - DS pseudo terminal.
4 - X.25 Packet Switching Network PAD (Packet Assembler Disassembler) terminal.

Refer to the MPE V Intrinsic Reference Manual (32033-90007) for more information.

FCLOSE Intrinsic Puts Three EOFs At the End of Unlabelled Tapes

By Gaetano J. Randazzo

The FCLOSE intrinsic has been enhanced for use with unlabelled magnetic tapes. If you use FCLOSE with a disposition of 0, 1, or 4 on an unlabelled magnetic tape, and the tape was written to while open, FCLOSE writes three End-Of-File marks at the end of the tape before performing a rewind or rewind/unload. This ensures that all tapes have an acceptable number of EOF marks at the end. The three EOFs are written only after the last FCLOSE to occur before the rewind, and only if the tape was written on.

This new procedure prevents an old problem with the file system: If only one EOF mark exists past the end of data on a tape and the tape is opened with two or more buffers, the file system will attempt to execute a preread past the single EOF marker. This is very annoying because the tape driver will perform a large number of retries to the tape. While the file system shields the user from these errors, the user still may think that something has gone wrong. Currently, these prereads cannot be avoided without significantly degrading performance.

Although new unlabelled tapes will end with three EOFs instead of one, old application programs will not be affected. Labelled tapes are not affected by this change. FCLOSE with disposition 3 (no rewind) works the same as before.

Three EOFs help the programmer as well: a programmer no longer has to write multiple EOFs to the tape with FCONTROL because the file system does it automatically. Programs that are already designed to write extra EOFs will not be affected because more than three EOFs at the end of a tape will still be recognized as End-Of-Data.

UTILITIES

SECTION

IV

SPOOK Tape Compatibility Solution

By Elizabeth Wei

The MPE utility program SPOOK enables spoolfiles to be transferred by means of SPOOK tapes. The SPOOK command >OUTPUT stores spoolfiles onto tape, and the >INPUT command brings spoolfiles into the system from tape. The format of the SPOOK tape includes system table entry images of the Device Class Table and Output Device Directory, both of which underwent expansion in MPE V/E. For this reason, SPOOK tapes from MPE IV and MPE V/E have different formats, although the spoolfiles themselves have not changed. Without special code to handle the foreign tape, a SPOOK tape could not be interpreted on a system running a different version of MPE than that which it was created on.

NOTE

All mentions of MPE IV also include MPE V/P.

To solve this compatibility problem two new versions of SPOOK, SPOOK4 and SPOOK5, have been introduced. The MPF V/E version of SPOOK (SPOOK5) has knowledge of MPE IV table formats, and can read MPE IV SPOOK tapes (as well as those of MPE V/E). Similarly, the new version of SPOOK running on MPE IV (SPOOK4) can read MPE V/E SPOOK tapes (as well as those of MPE IV). However, unlike reading SPOOK tapes, each version of SPOOK will only write SPOOK tapes of the same type as the MPE version it is running on. Checks are made to ensure that each version of SPOOK can only run on the appropriate version of MPE. For example, if an attempt is made to run SPOOK5 on an MPE IV system, SPOOK5 will abort with the message:

```
THIS VERSION OF THE UTILITY IS INCOMPATIBLE WITH THE CURRENT
MPE VERSION ON THIS SYSTEM.  (CIERR 1259)
```

An analogous message is printed for an attempt to run SPOOK4 on an MPE V/E system.

Both versions of SPOOK will be available in the MPE V/E (G.00.00) release, since they are required at that time by the MPE V/E user. A user wishing to transfer spoolfiles from an MPE V/E system to an MPE IV system would use SPOOK5 on the MPE V/E system to output, and SPOOK4 on an MPE IV system to input them. In order to ensure that MPE V/E SPOOK tapes can be read, updating to SPOOK4 from any previous version of SPOOK on the MPE IV system is necessary, since only SPOOK 4 understands MPE V/E SPOOK tapes. Similarly, MPE IV spoolfiles output by any MPE IV version of SPOOK can be input to an MPE V/E system using SPOOK5.

MPE V/E Utilities

By Bob Mead

The support of larger configurations with MPE V/E necessitated substantial changes to many of the data structures internal to MPE. As a result, most of the supported MPE utility programs required modifications in order to properly execute with MPE V/E. More importantly, in some cases, the execution of the MPE IV (and MPE V/P) version of an MPE utility on a system using MPE V/E may result in system hangs or system failures. To minimize potential problems due to accidental misuse of the wrong version of a utility, the names of the utilities that were modified have also been changed. The conversion program, PCONVERT, must be executed prior to updating with MPE V/E. The PCONVERT program will purge from PUB.SYS any copies of any MPE IV (and MPE V/P) utilities that could lead to failures under MPE V/E. Similarly, the backwards conversion process will purge the MPE V/E versions of the utilities. The complete list of names changes is given below, along with a list of utilities that did not require name changes.

We recognize that these name changes can create problems with UDCs and stream files that reference the old name of the utility. If the effort required to change UDCs, stream files, etc. is considered to be excessive, we recommend that the new versions of the utilities be copied into files with the previous names. By having copies of the utilities with both the old and new names on a system, users will be able to gradually convert to usage of the new names.

It should be noted that not only supported MPE utilities are affected by changes to MPE internal data structures. Many unsupported privileged utility programs will also cause system failures and hangs if they are not modified to reflect the MPE internals that changed with MPE V/E. Please call your SE if you have questions about whether your utility is supported or not supported.

The following supported MPE utilities did not require changes for MPE V/E:

PATCH SLPATCH MEMLOGAN MEMTIMER

The following supported utilities did require changes for MPE V/E, and will now have a new program file name:

MPE IV (V/P)	MPE V/E	REFERENCE
RECOVER2	RECOVER5	MPE V Utilities Reference Manual (32033-90008)
DPAN4	DPAN5	MPE V Utilities Reference Manual (32033-90008)
LISTLOG2	LISTLOG5	MPE V Utilities Reference Manual (32033-90008)
LISTEQ2	LISTEQ5	MPE V Utilities Reference Manual (32033-90008)
DISKED2	DISKED5	MPE V Utilities Reference Manual (32033-90008)
LISTDIR2	LISTDIR5	MPE V Utilities Reference Manual (32033-90008)
SPOOK	SPOOK5	MPE V Utilities Reference Manual (32033-90008)
ASOCTABL	ASOCTBL5	MPE V Utilities Reference Manual (32033-90008)
FREE2	FREE5	MPE V Utilities Reference Manual (32033-90008)
ATPDSM	TERMDSM	Terminal (DSM) On-Line Diagnostics Support Reference Manual (30144-90004)

SADUTIL, a stand-alone utility available as part of the Diagnostic Utility System (DUS), was also modified for MPE V/E. Its name was not changed however, due to the nature of the DUS tape. The MPE V/E version of DUS will function correctly on an MPE IV (and MPE V/P) system, but the MPE IV (and MPE V/P) version will not work under MPE V/E. If it becomes necessary to use SADUTIL, care should be taken to ensure that the MPE V/E DUS tape is used.

SADUTIL Enhanced, Works With Private Volumes

By Gaetano Randazzo

The Stand-Alone Diagnostic Utility (SADUTIL) now works with private volumes. This will be useful if the private volume directory or volume table becomes corrupt. To locate or recover files from a corrupt private volume, use the same SADUTIL commands as you would with a system volume. If the directory is intact but the volume table is destroyed, use the :SAVE command to store files on tape. If the directory is corrupt, use :FIND to scan the disc for specific files and store them on tape or serial disc.

Specify the Private Volume

To specify that you will be working with private volumes, respond "Y" when SADUTIL asks "PRIVATE VOLUME SET?" during the Configuration Dialogue. Configure the master disc of the private volume, instead of the system disc, as LDEV 1. Configure any slave private volumes as LDEVs 2, 3, etc. If the private volumes are configured in this way, SADUTIL can operate on the private volumes and not on system volumes. SADUTIL cannot work with private and system volumes at the same time.

FIND Accepts "Wild Card" File Names

The :FIND command, used to save files if the directory is corrupt, now accepts the wild card character specifier "@" in file set names. The :FIND command expects file sets to be in the form *filename .groupname .accountname* . Any of those parameters can be replaced by "@" to specify "all members of the set." This feature will save time when attempting to :FIND several files.

New Command CLID Resets Cold-Load IDs to 1

If INITIAL is aborted during a COOLSTART or WARMSTART, and some cold-load IDs have been updated and others have not, the next attempt to use INITIAL will report that the volume table is corrupt and that a RELOAD is necessary. Actually, the only problem is that the cold-load IDs do not match. In that case, use the CLID command to reset the cold-load IDs to 1, to avoid performing a RELOAD.

HELP Command is New

The new :HELP command offers an explanation of all SADUTIL commands.

SADUTIL Sample Dialog

The sample dialog that follows demonstrates the the new features of SADUTIL. For details, refer to the MPE V Utilities Reference Manual (32033-90008).

```
Diagnostic/Utility System Revision 02.11
Enter Your Program Name (type HELP for program information)
:SADUTIL
Disc Utility   03.03 (C) Hewlett-Packard Co., 1982

<<CONFIGURE ALL DISKS IN VOLUME SET>>

PRIVATE VOLUME SET? Y
LOGICAL DEVICE? 1
DRT? 49
UNIT? 0
TYPE? 0
SUBTYPE? 8
LOGICAL DEVICE? 2   ** Configure all discs in the current volume set **
DRT? 49
UNIT? 1
TYPE? 0
SUBTYPE? 8
LOGICAL DEVICE?  RETURN
LIST LOGICAL DEVICES? Y

 LDEV  DRT  UNIT  TYPE  SUBTYPE
 ----  ---  ----  ----  -------

   1    49    0     0      8
   2    49    1     0      8

SERIAL DEVICE CHANGES? Y     ** Configure Serial Device **
DRT? 41
UNIT? 0
TYPE? 24
SUBTYPE? 0
LIST SERIAL DEVICE? Y        ** Serial devices are now allowed **

 LDEV  DRT  UNIT  TYPE  SUBTYPE
 ----  ---  ----  ----  -------

   0    41    0    24      0

ENTER FUNCTION: HELP         ** New HELP command **

 Below are all the commands supported by SADUTIL:

CLID                     Set all Cold Load IDs to 1.
CONF [ldev]              Configure logical devices.
COPY                     Copy one disk to another.
DBUG                     Enter symbolic debugger.
EDIT                     Below are the five edit commands.
  BASE [basesector]      Specifies base sector number.
  DISC [ldev]            Specifies logical device to edit.
  MODIFY [diskaddr][,wordloc][,wordcount] Modify a sector.
  PDSK                   Print Disk. Enter address as
                         [address][,sectcount][;A!O].
  OUTM [C!P]             Output to Console or Printer.
```

```
FIND                  Scan a disk for file labels.
HELP                  Explain facility.
OUTM [C!P]            Output to Console or Printer.
PDTT [ldev]           Print Defective Tracks Table.
PDSK [ldev]           Print Disk. Enter address as
                      [address][,sectcount][;A!O].
PFIL                  Print file names from directory.
PVOL [ldev]           Print Volume Label of ldev.
SAVE                  Save files to serial device.
STOP                  Exit SADUTIL.

ENTER FUNCTION: FIND                ** FIND command showing new feature **
FIND scans for file labels.
There is no guarantee that the labels or files are intact.
If the files are requested to be saved many may be bad or already purged.
IF the file info ends with '?????', then the file is either already purged
     or it was left open when the system failed.
Files listed with '?????' will not be saved.
An attempt can be made to save via SAVE by LDEV# and sector address.

Do you wish files found to be saved? Y
Enter LDEV number to scan: 1
Enter file set to find: @.RANDAZZO.MPEM    ** File subset specific **
DATE ?  1/1/83

                .
                .
     ** FIND command output **
                .
                .

ENTER FUNCTION: CLID         ** New CLID command **

WARNING!! This function will rewrite all Cold Load IDs.
Are all system domain volumes mounted and ready? Y

Cold load IDs written in system tables.
Cold load ID written on LDEV#1.
Cold load ID written on LDEV#2.

ENTER FUNCTION: STOP
```

Changes That Affect Contributed Utilities

By Ben Lloyd, Technical Manager, Interex

When is a bug not a bug? When it is the result of an enhancement. The HPIUG Contributed Software Library (CSL) has long been an alternate source of system utilities, supplementing those supported by Hewlett-Packard. Over the years, many of these utilities have been incorporated into new releases of MPE, making certain Library contributions obsolete. At the same time, substantial internal changes have caused many of the more operating system dependent CSL utilities to become inoperative. In the case of MPE V/E, several of the Old Guard are affected.

In the case of contributions whose functions have been incorporated into MPE, RESTORE, GETFILE, JOBMAIN, FETCHIT, PARITY, and SESID, they are no longer needed. The functions of RESTORE, GETFILE, FETCHIT and PARITY have been included in the newest version of MPE RESTORE, while the intrinsic JOBINFO provides all the information gleaned through the use of JOBMAIN and SESID. Good job!

On the other side of the coin, there are several changes to system tables that have a substantial and detrimental effect on the operation of several privileged CSL utilities. In general, anything that reads system tables will have to be updated before it will work properly. In evaluating whether to use a contribution on your system, you must determine whether or not the contribution attempts to make modifications of system tables, or simply reads them. In the former case, no attempt should be made to run the contribution. In the latter, you may run it, but any information retrieved should be carefully scrutinized. In the following discussion, I have attempted to distinguish those contributions that try to modify system tables, and have introduced a preceding asterisk to indicate them.

There are several categories of changes that affect these contributions: The first involves the substantial changes made to the Logical Device Table (LDT). This change affects *BOUNCER, LOSTDISC, SPOOLTRM, MOO, TUNER, and TUNER4 from Release A0. Obviously, related contributions from earlier releases will also be affected (WATCHDOG, SOO, etc).

The second category results from the expansion of the LDEV from 1 byte to 1 word, necessitating smaller changes in a large number of tables. Affected tables include the DIT, LDT, ODD, IDD, DCT, etc, which can have a substantial impact on a sizable number of contributions. Included in this category are *JSPOOK, SETUP, SHOWVM, *SLAMMER, *SPOOLZAP, *TDELAY, TUNER, and TUNER4 from Release A0. Any contribution that evaluates a table with a nondisc LDEV (the disc LDEV# is still limited to 8 bits (#256)) will not work.

Next in line are contributions that examine PCB entries. The PIN has been expanded, and the PCB table has been shuffled around, translating to all PCB related tables (PCBX, JMAT, etc). Into this pool fall such wonders as *XCALIBUR, MOO, TUNER, TUNER4, *LIMCHNG, *SETJF, SYSINFO, and *ABORTJOB.

Along similar lines comes PARMINFO. With MPE V/E, the change in the stack marker has made tracing through the stack markers (in privileged mode) to retrieve the run-time PARM or INFO string a little bit trickier. The current version of PARMINFO is not tricky enough.

The final two table changes result from the expansion of the DST and CST tables. The new CST table has been broken into two parts, affecting all examinations of the CST#. The DST# has been expanded to allow values up to 4096. These changes affect any utilities that examine memory management tables or use the MFDS and MTDS instructions with restricted length DST#s. This has its effect on virtually all of the contributions in the TECH account.

The third type of change is in the System Log format. Due to the expansion of various items (PIN, LDEV, etc), the system log file entries were changed. This has an impact on all contributions that read/evaluate the log files (system only, not user log files). Included here are nonprivileged utilities such as LOGUTIL, CONSLOG, LOGREPT, etc. If you use any of these contributions, your data results will be skewed when analyzing log entries of the following types: 4 (Process Termination), 5 (File Closes), 11 (I/O Errors), 13 (Logical Mounts/Dismounts), 14 (Tape Labels), 17 (Call Progress Signals), and 18 (DCE Provided Info).

All contributions that are affected by these changes will be updated in the next release of the Library (Release B0), but will not work in their current form (from any prior release; e.g. Release A0, A1, 09, etc). The Users Group may be able to produce an MPE V/E Release Tape this summer, news of which will be distributed in INTERRUPT. Keep your eyes open for news thereof.

"PEP" Up Disc Performance

By Kent Wilken and Kwei-Ping Wong

The Performance Enhancements Project, or PEP, is a firmware upgrade for all CS-80 products supported on the HP 3000: HP 7911, 7912, 7914, 7933, and 7935. PEP has improved performance by redesigning the firmware to run more efficiently. The upgrade is currently available at no cost to the customer, if installed on the next normal CE visit.

For a system with a single HP 7933 disc drive, our testing has shown that a typical improvement with PEP is between 15 and 20 percent; the improvement may be more or less than this depending on the application. This improvement in performance is the result of a 63% decrease in controller overhead.

In addition to streamlining the firmware, a new feature, called Buffer Prefill, was added to the CS-80 disc controller. When many disc drives are connected to the same HP-IB channel, one disc drive may be ready to do a transfer while a second is using the HP-IB channel. Before PEP, the first drive would not do the transfer. It would be forced to do a full rotation (22 ms on an HP 7933) before a transfer over the HP-IB channel was again attempted. With the new Buffer Prefill feature, if a drive is blocked from accessing the HP-IB channel, it will do a transfer from the disc to its internal 4 Kbyte buffer. (This only happens if the entire transfer will fit in the buffer.) As soon as the HP-IB channel is freed, the data can be transferred immediately from the disc buffer, avoiding the added rotational delay. With four HP 7933 discs on the same Series 44 HP-IB channel, this feature alone can provide up to a 40% improvement in disc I/O throughput.

In conjunction with the PEP improvements in the disc, the version of the CS-80 driver released with MPE V/E has new performance features:

- The CS-80 disc driver has been tuned for improved performance, resulting in driver execution time being reduced by 15%.
- Rotational Position Sensing (RPS) is automatically enabled by the new driver for all CS-80 discs with PEP firmware. RPS is a method of more effectively using the HP-IB channel when multiple drives are active on the HP-IB channel. With RPS, the drive will signal its availability to do an I/O only when it is a small rotational distance (position) away from the target data, hence the name Rotational Position Sensing.

RPS maximizes HP-IB channel usage by not connecting drives to the HP-IB channel that are a great distance away from the target data. With four HP 7933 discs on the same Series 44 HP-IB channel, RPS can provide up to a 20% improvement in I/O throughput beyond the improvements found in PEP.

The combined improvement of all the PEP and driver enhancements has increased the throughput of four CS-80 drives on an HP-IB channel by as much as 90% when running a synthetic, disc I/O intensive, application.

RPS should improve performance on all systems with multiple master disc controllers on the same HP-IB channel, assuming the drives are candidates for enabling RPS (i.e., have PEP firmware). On single disc systems there is no need for RPS, since there is only one unit requesting HP-IB channel service. There may even be a slight performance degradation due to RPS controller overhead. This may also be the case, when a mixture of HP 7920/25 and CS-80 drives share the same HP-IB channel. For these, and other configurations or applications, a new command has been added to MPE V/E. The :DISCRPS command alters the status of RPS. This command takes effect immediately without the need to restart the system.

To execute the :DISCRPS command you must be one of the following:

- Master operator at the console.
- Associated with the device class through the :ASSOCIATE command.
- Allowed to use :DISCRPS through the :ALLOW command.

The syntax is:

```
                      [,ENABLE ]
:DISCRPS ldev         [,DISABLE]
```

Also, the command :SHOWDEV has been modified to indicate those CS-80 discs in which RPS has been enabled. For example:

```
:SHOWDEV

LDEV        AVAIL            OWNERSHIP     VOID    DEN  ASSOCIATION
1           DISC    (RPS)    50 FILES
```

PEP has been shipping since the beginning of last May for HP 7933/35, and since the beginning of last June for HP 7911/12/14.

Double Buffering and Faster Gap Table Searches in MPE V/E Serial Disc

By Larry Byler

Two enhancements have been added to the MPE Serial Disc facility (SDISC), starting with MPE V/E: a second internal buffer and a faster Gap Table search algorithm.

Double Buffering

In earlier versions of MPE, SDISC had one internal data buffer. The length was 32 Kb when a cartridge tape was used and 8 Kb for all other serial disc devices. User file system calls resulted in data movement between the user's buffer (or File System buffer) and SDISC's buffer. Only when SDISC's buffer was full (writing) or empty (reading) was physical I/O required. Because SDISC used only one buffer, this I/O had to be done with WAIT. This type of operation was acceptable before the cartridge tape was released, because even with WAITed I/O, the serial disc was substantially faster than the process trying to fill or empty its buffer. (For the rest of this section, we will limit the description to writing, and trying to keep the SDISC buffer(s) full.)

When the serial disc device could be a cartridge tape, the process became serial disc bound, and the larger buffer required by the cartridge tape also took longer for the process to refill when the buffer became available again. The result was inferior performance.

The MPE lab ran some average transfer rate studies, using :STORE with several typical file mixes. Observed transfer rates were between 0.7 and 1.2 Mb/minute, depending primarily on the file sizes. (Shorter files, which required more MPE directory and File System overhead, took longer to transfer than did longer files.) If the cartridge tape moved continuously, it could transfer data at 2.1 Mb/minute.

Investigation showed that a major effort would be required to keep the cartridge tape moving continuously. However, by installing a second SDISC (MPE Serial Disc facility) buffer and expanding the buffer sizes to the maximum allowed by MPE (just under 64 Kb), the transfer rate would improve to a predicted 1.79 Mb/minute (85% of the 2.1 Mb/minute maximum) independent of the file size. Accordingly, SDISC now has two data buffers which allow physical I/O without WAIT. That is, one buffer is available to the user (via the File System) while the device is servicing the other.

Both buffers are allocated and locked in memory when the device is first opened. They are returned to MPE when the device is closed. If the system does not have two 64 Kb areas available, SDISC tries to get smaller buffers (in 8 Kb increments). This continues until the buffer lengths reach 8 Kb. SDISC then tries for a single buffer in the same way. If one 8 Kb area cannot be found, SDISC reports an error and will not continue. Naturally, the transfer rate is degraded each time SDISC must settle for less than the maximum size buffers. If SDISC has to back all the way down to a single buffer (extremely unlikely) we are back to pre-MPE V/E performance, since all physical I/O must again be done with WAIT.

Measurements using two 64 Kb buffers indicate a transfer rate of 1.67 Mb/minute, or 93% of the predicted rate.

Faster Gap Table Search

Double buffering enhances both read and write SDISC performance. The fast Gap Table search only improves reading, since the Gap Table is not searched while SDISC is writing.

The Gap Table is an SDISC internal data structure. One of its several uses is to contain End of File (EOF) address entries, that is, device addresses of a logical EOF. Skipping over a file is then as easy as scanning the Gap Table; there is no need to read and discard the actual data. Each file has a corresponding EOF entry, so many short files mean many entries, and higher capacity discs also mean more entries.

Because of other SDISC constraints, the Gap Table had to be completely scanned twice each time SDISC asked for it. In addition, SDISC asked for it two or three times per FREAD request. Before the HP 7933/7935 discs, this inefficiency was not noticed because the Gap Table was quite small.

The present enhancement reduces the scanning time by restricting the scan to a narrow area of interest within the Gap Table, and this area must only be scanned once. To make this possible, the following changes have been made to SDISC:

- "Hole" entries have been eliminated from the Gap Table.
- Conditions that led to the creation of hole entries, namely ATTACHIO errors, have been minimized by reassigning suspect or deleted tracks in the Defective Tracks Table (HP 7905/06/20/25 discs) or Defective Sector Table (CS-80 discs) whenever writing starts. SDISC restricts this operation to tracks beyond the current location, so as not to disturb previously-written data.
- To maintain compatibility with existing serial discs, the former Gap Table scanning algorithm has been retained and will be used on these older packs. An internal version code enables SDISC to tell the difference between pre- and post-MPE V/E serial discs.
- Older packs can be converted to the MPE V/E version by the VINIT >SERIAL command of MPE V/E. Naturally, this erases all data on the disc; make sure there is nothing of value on it. In addition, SDISC will perform the conversion automatically whenever writing starts at the Load Point, as is the case with all :SYSDUMP and :STORE operations.

Directory Expanded?

By Don Darnell and Rob Fisher

To ensure that the installation of MPE V/E will be successful, customers must verify one of two items:

- Check to see if the latest version of the directory expansion patch is installed on the system. You may verify this with your SE.

or

- If the latest version of the directory expansion patch has not been installed, then verify that the directory size is less than 6000 sectors.

To verify that the directory size is less than 6000 sectors, follow this procedure:

1. The current size of the directory can be verified by performing a "SYSDUMP $NULL " When the question "ANY CHANGES? ", is displayed, enter:

   ```
   YES
   ```

2. Press RETURN for each additional question until "DISC ALLOCATION CHANGES? " is displayed. Enter:

   ```
   YES
   ```

   SYSDUMP will output:

   ```
   DIRECTORY USED = xxxx, MIN = yyyyy, MAX = zzzz.?
   ```

3. If the MAX=*zzzz* value is more than 6000 sectors, do not update to MPE V/E. If you have any questions please call your SE or SEO (Service Engineering Organization). If the value is 6,000 sectors or less, the customer can proceed to update to MPE V/E.

Code Segment Table Expansion in MPE V/E

By Jessy J. Hsu

The Code Segment Table (CST) has been expanded in MPE V/E. A system running with new microcode and MPE V/E can have up to 2048 CST entries. A single process can contain segments up to 255, in any combination of program segments and user Segmented Library (SL) segments. Several changes have been made to accommodate the CST enhancements. These changes do not apply to those systems still running old microcode under MPE V/E. Please see your System Manager if you have a question about the version of microcode.

CST Changes

The CST is now partitioned into two parts. The first part consists of 192 to 255 entries and is reserved for MPE segments. It is called the physical domain CST. The second part of the CST is called the logical domain CST. The maximum physical domain CST entry number is not configurable, while the maximum logical domain CST number can be configured.

Program Loading Mechanism

A program having *n* segments and referencing user SL segments, will be loaded in the following manner. The *n* program segments are loaded into the Code Segment Table Expansion (CSTX), as they were in MPE IV and MPE V/P. The SL segments are loaded into the logical domain CST. Segment numbers for program segments start at 1. Segment numbers for SL segments start at *n*+1; both are logical segments to this particular process that is executing the program. The logical segment numbers are found in the external P-Label or Status Register of the Stack Marker. The MPE V/E Loader will create a data segment for mapping these segment numbers into physical numbers. The microcode will access this map and obtain the physical CST.

P-Label and Stack Marker Changes

A flag is needed to indicate whether a segment number in an external P-Label or a stack marker is a physical or logical segment. The P-Label and Stack marker are changed to the following form:

```
          External P-Label:
----------------------------------
|m|     STT        | Segment Number |
----------------------------------
 0                               15
              Stack Marker:
----------------------------------
|              X Register          |
|----------------------------------|
|  |m|           Delta-P           |
|----------------------------------|
|                  | Segment Number |  => Status Register
|----------------------------------|
|                Delta-Q           |
----------------------------------
 0                               15
```

The *m* is a mapping flag. If *m* = 1, a physical segment is indicated, and if *m* = 0, a logical segment is indicated .

The head of the Segment Transfer Table (STT) is also changed such that the microcode can distinguish internal P-Labels from external ones as follows:

```
-----------------------------------
| # internal STTs| # Total STTs    |
-----------------------------------
```

PCAL Instruction Change

The internal label case of PCAL 0 has been eliminated because of the change in the definition of P-Label bit (0:1). The MPE V/E microcode cannot determine whether the label on the top of stack is an internal or external label.

Compatibility Considerations

Compiled programs can be transported back and forth between MPE IV, MPE V/P, and MPE V/E without recompiling. Loaders in MPE IV, MPE V/P, and MPE V/E are able to convert the Segment Transfer Table (STT) to the proper form for both program and SL segments.

Segmented Library File Segment Limit Change

Due to the new loading mechanism, the maximum segment number in an SL file has been changed to 254. In MPE IV and MPE V/P, it is 255.

Walking Back Through Stack Markers

To locate the INFO or PARM specified in the :RUN command, user code can walk back through stack markers to find the bottom marker. The user code does this by looking for the Delta-P word of the stack marker, where bits (2:14) are set to zero. In MPE V/E, do not depend on bits (0:2) in the Delta-P word of the terminating marker to be equal to zero.

Impact of CST Expansion on Compiled Basic Programs

by Bob Mead

A change required to support expansion of the CST table was elimination of the internal label option of the PCAL 0 instruction. The PCAL 0 construct allowed a PLABEL to be loaded onto the Top Of Stack (TOS), then executed directly, and usually called external procedures (i.e., procedures outside the current code segment). The only known case where a PCAL 0 to an internal label is used is in the implementation of user-defined functions in compiled BASIC programs. Provisions were made in the MPE V/E firmware for this construct, based on the assumption the PCAL 0 (local label) was always immediately preceded by an LRA instruction (used to load the label on TOS).

In actuality, the LRA and PCAL 0 instructions are generated in logically adjacent code locations, but the BASIC compiler may insert constants between the two instructions, resulting in the following sequence:

```
LRA P+2,I
BR  P+3
relativeaddress      **Target of PCAL**
constant
PCAL 0
```

Unfortunately, the provisions for a PCAL 0 to an internal label in the MPE V/E firmware are incapable of handling this particular code sequence, and the program will abort with either a CST violation or STT violation.

As a result, once MPE V/E firmware is installed on a system some compiled BASIC programs may need to be recompiled, or they may abort when executed. The BASIC compiler (B.0.20) has been modified to ensure that the constants will not be inserted between the LRA and PCAL 0 instructions, and thus once recompiled, a program should no longer abort. To facilitate identifying BASIC programs that may require recompilation, a utility has been provided that will indicate whether or not a BASIC program should be recompiled. This utility, PVERIFY.PUB.SYS , prompts the user for the name of a program file, and then analyzes the code contained in the program file to determine whether recompilation is necessary. As an example, suppose we have compiled BASIC programs BAD and GOOD . The following should be done to determine if recompilation is necessary:

```
:RUN PVERIFY.PUB.SYS

MPE V Program Verifier V.00.00 (C) Hewlett-Packard Co. 1983
Enter file name to be verified
Or enter // to exit
? BAD.PUB.SYS

Program file         : BAD.PUB.SYS
# Suspected PCAL 0   : 2
Recompile required   : Yes

Enter file name to be verified
Or enter // to exit
? GOOD.PUB.SYS
```

```
Program file          : GOOD.PUB.SYS
# Suspected PCAL 0    : None
Recompile required    : No

Enter file name to be verified
OR enter // to exit
? //

END OF PROGRAM
```

In the above example, it would be necessary to recompile the program file BAD , but the program file GOOD would not require a recompilation.

If the source for a program that requires recompilation is not available, it is possible to patch the program file directly using the utility PATCH.PUB.SYS . In the event this is necessary, the PVERIFY utility has an entry point, LISTCODE, that will print out the locations of the instructions involved in the required patch. As an example:

```
:RUN PVERIFY.PUB.SYS,LISTCODE

MPE V Program Verifier V.00.00 (C) Hewlett-Packard Co. 1983
Enter file name to be verified
Or enter // to exit
? NOSRC.PUB.SYS

Segment    PB rel. address    Code
-------    ---------------    ----

000001         001102        LRA  P+3,I
               001103        BR   P+3
               001106        PCAL 0

000034         002767        LRA  P+3,I
               002770        BR   P+3
               002773        PCAL 0

Program file          : NOSRC.PUB.SYS
# Suspected PCAL 0    : 2
Recompile required    : Yes
```

If patching of a program file is required, we recommend that you contact your SE prior to performing the patch, to verify that the intended patch is correct. The patch must guarantee that the LRA and PCAL 0 instructions are physically adjacent. If not, at least one instruction and one constant must be moved. Movement of the constant will require changes to any instructions referencing the constant. Recalculation of the constant in the indirect cell for the LRA instruction is also required.

File System Buffer Size Increased

By Gaetano J. Randazzo

The total amount of file system buffering space has been increased by approximately 18K words. Previously, the maximum space utilized for buffers by the file system was 14K words. This limit has been increased to 32K, the maximum extra data segment size allowed by MPE.

File system buffers are used to store physical file records. Each physical record, or block, is made up of 1 or more logical records. When a file is FOPENed as buffered, the file system performs I/O to the device to which the file is associated (disc, tape, etc) in sizes of one block. This block is kept in file system buffers in memory and data is transferred between the user's buffer and the file system buffer one logical record at a time via file system intrinsics.

With the old restriction, if the block size of a file was greater than 14000 words, file system buffering could not be performed. This meant that the user had to perform his own blocking and deblocking of the file's logical records and the file had to be opened NOBUF. The data transfer of blocks was performed directly between the user's buffer and the device. If the block size was less than 14K and greater than 7K, then only one file system buffer could be obtained. It is important to obtain at least two file system buffers because I/O from one of the buffers can be completing while the other buffer is being blocked or deblocked with logical records. It is not usually advantageous to open a file with more than two to four buffers. In most cases, the I/O from any one buffer is complete before it is needed again. With this new increase in buffer space, the user can now open files with larger block sizes and still be able to fully utilize file system buffering. The advantage of placing more logical records per physical record, thus having larger blocks, is that the number of I/Os to the device is decreased. This is always very important because I/O access is many times slower than direct access from buffers kept in memory.

The table below demonstrates the importance of having large data transfers to devices. The table does not claim to characterize file system buffering performance; it is only intended to demonstrate that large buffers improve performance.

#	TYPE	RSIZE	R/B	BLK	BUFF	NUM	R/W	REAL	CPU
1	TAPE	1024	1	1024	2	1000	W	56.356	5.385
2	TAPE	1024	8	8192	2	1000	W	31.781	2.884
3	DISC	1024	1	1024	2	1000	W	21.827	7.404
4	DISC	1024	8	8192	1	1000	W	13.471	5.756
5	DISC	1024	8	8192	2	1000	W	7.275	5.883

#	-	Test number.
TYPE	-	Type of file being accessed, either tape or disc.
RSIZE	-	Record size in words.
R/B	-	Number of records per block.
BLK	-	Block size or physical record size in words.

BUFF - Number of file system buffers.
NUM - Number of transactions performed.
R/W - Read or write to the file.
REAL - Transaction time in real time (wall-clock) seconds.
CPU - The number of CPU seconds used.

As the table shows, a substantial increase in performance can be obtained when the block size of the file is increased. Especially note the difference between test 4 and test 5 in the number of buffers. Test 4 shows the default buffering for MPE IV with an 8K block size. Due to the limitation, only one buffer could be obtained. However, in test 5 (MPE V/E), up to two buffers were obtained, increasing the performance dramatically for this size of the block. Real time for this block size was almost cut in half. Note also the difference between (test 1 and 2) and (test 3 and 4). Increasing the blocking factor from 1 to 8 (and the physical block size), both the real and CPU times were greatly reduced. These types of data transfers and files will be able to take full advantage of the increased file system buffer size.

System Logging Changes for MPE V/E

By Jan Helmbolt

Some changes have been made to system logging record formats for MPE V/E. Most of these changes were necessitated by the new limits of the MPE table structures. For example, the PIN of a process cannot be held in 8 bits any longer, and thus the logging records have been changed to handle the larger PINs and LDEVs.

In addition, one enhancement has been added to the Process Termination logging record. The process' PIN and CPU time are now indicated in the Type 4 record.

The utility LISTLOG2.PUB.SYS does not exist on MPE V/E; it has been replaced by the utility LISTLOG5.PUB.SYS. This utility handles all logging changes.

The following charts show the new logging record formats:

```
Field Length                        Values Always
     (Words)    Contents                    Taken
        |-----------------------------------|
     (1)|        Record Type                |  =  4
        |-----------------------------------|
     (1)|        Record Length              |  = 14
        |-----------------------------------|
     (3)|                                   |
        |            Time Stamp             |
        |                                   |
        |-----------------------------------|
     (1)|0    1|                          15|
        | job  |                            |
        | type|     Job Number              |
        |-----------------------------------|
     (1)|      # Of Program File Segments   |
        |-----------------------------------|
     (1)| # Of SL Segments (Non MPE)        |
        |-----------------------------------|
     (1)|      Maximum Stack Size Ever      |
        |-----------------------------------|
     (1)|  Maximum Data Segment Size Ever   |
        |-----------------------------------|
     (1)|Cumulative Total of Virtual Storage|
        |-----------------------------------|
     (1)|       PIN of Process              |
        |-----------------------------------|
     (2)|       CPU Time in Seconds         |
        |-----------------------------------|
        Process Termination Record (Type 4)
```

```
Field Length                    Values Always
     (Words)    Contents                Taken
        |-----------------------------------|
     (1)|       Record Type                 |  =  5
        |-----------------------------------|
     (1)|       Record Length               |  = 29
        |-----------------------------------|
     (3)|                                   |
        |           Time Stamp              |
        |                                   |
        |-----------------------------------|
     (1)|0   1|                           15|
        | job |                             |
        | type|     Job Number              |
        |-----------------------------------|
    (14)|                                   |
        |                                   |
        |                                   |
        |                                   |
        |                                   |
        |                                   |
        |           File Name               |
        |                                   |
        |                                   |
        |                                   |
        |                                   |
        |                                   |
        |                 |-----------------|
        |                 |  (reserved )    |
        |-----------------------------------|
     (1)|0 Disposition   7|8    Domain    15|
        |-----------------------------------|
     (2)|                                   |
        |        # of Sector Allocated      |
        |-----------------------------------|
    r(1)|0 Device type   7|/////////////////|
        |-----------------------------------|
     (2)|                                   |
        |       # of Records Processed      |
        |-----------------------------------|
     (2)|                                   |
        |       # of Blocks Processed       |
        |-----------------------------------|
     (1)|          Device Number            |
        |-----------------------------------|
        File Close Record (Type 5)
```

```
Field Length                         Values Always
     (Words)    Contents                     Taken
        |-----------------------------------|
     (1)|       Record Type                 |  = 11
        |-----------------------------------|
     (1)|       Record Length               |  = 21 + Status Words Logged
        |-----------------------------------|
     (3)|                                   |
        |           Time Stamp              |
        |                                   |
        |-----------------------------------|
     (1)|0    1|                          15|
        | job  |                            |
        | type|      Job Number             |
        |-----------------------------------|
     (1)|     Error Index | Word*           |
        |-----------------------------------|
     (1)| Flags Words (see sys mgr/sys sup.)|
        |-----------------------------------|
     (1)|     Queue LDEV Number             |
        |-----------------------------------|
     (1)|Miscellaneous (Driver Defined) Data|
        |-----------------------------------|
     (1)|   Target Data Segment             |
        |-----------------------------------|
     (1)|   Target Data Address             |
        |-----------------------------------|
     (1)|     Unit         |    Function    |
        |-----------------------------------|
     (1)|  Transmission Count (Neg if bytes)|
        |-----------------------------------|
     (1)|   Driver Defined Parameter = 1    |
        |-----------------------------------|
     (1)|   Driver Defined Parameter = 1    |
        |-----------------------------------|
     (1)|//////////////////| Stat    |1 Stat|
        |//////////////////|Qualifier |3     |
        |-----------------------------------|
     (1)|    Process PCB  Number            |
        |-----------------------------------|
     (1)|        |subtype |  Device Type     |
        |-----------------------------------|
     (1)|      Hardware Unit Number         |
        |-----------------------------------|
     (1)|          DRT Number               |
        |-----------------------------------|
     (n)|       Status Words Logged         |
        |-----------------------------------|
        I/O Error Record (Type 11)
```

```
Field Length                          Values Always
     (Words)   Contents                       Taken
        |-----------------------------------|
     (1)|        Record Type                |  = 13
        |-----------------------------------|
     (1)|        Record Length              |  = 41
        |-----------------------------------|
     (3)|                                   |
        |            Time Stamp             |
        |                                   |
        |-----------------------------------|
     (1)|0   1|                           15|
        | type|      Job Number             |
        | job |                             |
        |-----------------------------------|
     (1)| No. of Accessors  11|12 TR 14|MD  |
        |-----------------------------------|
     (1)|         PIN of Accessor           |
        |-----------------------------------|
     (4)|                                   |
        |                                   |
        |                                   |
        |           User Name               |
        |-----------------------------------|
     (4)|                                   |
        |                                   |
        |           Group Name              |
        |                                   |
        |-----------------------------------|
     (4)|                                   |
        |                                   |
        |          Account Name             |
        |                                   |
        |-----------------------------------|
     (4)|                                   |
        |                                   |
        |          Volume Set Name          |
        |                                   |
        |-----------------------------------|
     (4)|                                   |
        |                                   |
        |          Volume Set Group         |
        |                                   |
        |-----------------------------------|
     (4)|                                   |
        |                                   |
        |          Volume Set Account       |
        |                                   |
        |-----------------------------------|
     (1)| No. Volumes Mounted/Dismounted    |
        |-----------------------------------|
     (1)|0 Device Subtype |  1st Volume LDEV|
        |-----------------------------------|
     (1)|  Device Subtype |  2nd Volume LDEV|
        |-----------------------------------|
     (1)|  Device Subtype |  3rd Volume LDEV|
        |-----------------------------------|
     (1)|  Device Subtype |  4th Volume LDEV|
        |-----------------------------------|
     (1)|  Device Subtype |  5th Volume LDEV|
        |-----------------------------------|
     (1)|  Device Subtype |  6th Volume LDEV|
        |-----------------------------------|
     (1)|  Device Subtype |  7th Volume LDEV|
        |-----------------------------------|
     (1)|  Device Subtype |  8th Volume LDEV|
        |-----------------------------------|
        Logical Mount/Dismount Record (Type 13)

Field Length                    Values Always
     (Words)    Contents                Taken
        |-----------------------------------|
     (1)|       Record Type                 |  = 14
        |-----------------------------------|
     (1)|       Record Length               |  = 31
        |-----------------------------------|
     (3)|                                   |
        |           Time Stamp              |
        |                                   |
        |-----------------------------------|
     (1)|0   1|                           15|
        | job |                             |
        | type|     Job Number              |
        |-----------------------------------|
     (1)|         Logical Device Number     |
        |-----------------------------------|
     (1)|0F |/|2    File Sequence Number    |
        | T |/|                             |
        |-----------------------------------|
     (1)|0  File Number  7|Seq|type|T|A|L|W |  Seq type = 2 bits
        |                 |typ|    |M| | |  |      type = 2 bits
        |-----------------------------------|        TM = 2 bits
     (1)|/////////////////| Volume Seq. Num.|     A,L,W = 1  bits
        |-----------------------------------|
     (1)|      Expiration Date              |
        |-----------------------------------|
     (9)|                                   |
        |                                   |
        |                                   |
        |                                   |
        |           File Name               |
        |                                   |
        |                                   |
        |                                   |
        |                 |   reserved      |
        |-----------------------------------|
     (4)|                                   |
        |                                   |
        |           Lockword                 |
        |                                    |
        |------------------------------------|
     (3)|                                    |
        |        Volume Set ID               |
        |                                    |
        |------------------------------------|
     (3)|                                    |
        |           Volume ID                |
        |                                    |
        |------------------------------------|
     (1)|           Pin Number               |
        |------------------------------------|
        Tape Labels Log Record (Type 14)

Field Length                     Values Always
     (Words)    Contents                 Taken
        |------------------------------------|
     (1)|       Record Type                  |  = 17
        |------------------------------------|
     (1)|       Record Length                |  =  8
        |------------------------------------|
     (3)|                                    |
        |           Time Stamp               |
        |                                    |
        |------------------------------------|
     (1)|0   1|                            15|
        | job |
        | type|     Job Number               |
        |------------------------------------|
     (1)|        CPS Message Number          |
        |------------------------------------|
     (1)|           DS LDEV Number           |
        |------------------------------------|
        Call Progress Signals Log Record Format (Type 17)

Field Length                     Values Always
     (Words)    Contents                 Taken
        |------------------------------------|
     (1)|       Record Type                  |  = 18
        |------------------------------------|
     (1)|       Record Length                |  = 10
        |------------------------------------|
     (3)|                                    |
        |           Time Stamp               |
        |                                    |
        |------------------------------------|
     (1)|0   1|                            15|
        | type|     Job Number               |
        | job |                              |
        |------------------------------------|
     (1)|        CPS Message Number          |
        |------------------------------------|
     (1)|           DS LDEV number           |
   |-----------------------------------|
(2)|                                   |
   |            ***                    |
   |-----------------------------------|
   DCE Provided Info (Type 18)
```

The following is an explanation of the Message Numbers found in DCE Provided Info (Type 18):

```
           ***   Message Number
   |-----------------------------------|
   | 0|                                |
   |--| Monitory Charge (no fraction)  |
   |                                   |
   |-----------------------------------|
       1 (charge information type 1)

   |-----------------------------------|
   | 1| fraction     |                 |
   |----- ----------|                  |
   |    Monitory Charge (w/ fraction)  |
   |-----------------------------------|
     2 (charge information type 2)

   |-----------------------------------|
(8)|      Time (seconds)               |
(9)|                                   |
   |-----------------------------------|
       3(charge information type 3)

   |-----------------------------------|
(8)|            Units                  |
(9)|                                   |
   |-----------------------------------|
     10 (line identification signal)

   |---|-------------------------------|
(8)| 0 |                               |
   |---| International Data Number     |
(9)|                                   |
   |-----------------------------------|

   |---|-------------------------------|
(8)| 1 | National Number  or           |
   |---|                               |
(9)|     Network Terminal Number       |
   |-----------------------------------|
```

INITIAL/SYSDUMP Dialog Changes With This Release

By Donna Sidman

The MPE Initiator and SYSDUMP Configurator dialogs have been slightly changed with the G.00.00 release of MPE in order to accomodate the Workstation Configurator software (refer to "DSN/WORKSTATION CONFIGURATOR"). To take advantage of this software, you must first (prior to running the dialogs) define your own terminal types and save the terminal type information in a disc file, called a descriptor file, with TTUTIL.PUB.SYS of the Workstation Configurator. (For more information, refer to the Workstation Configurator Reference Manual (30239-90001).)

If the device type is 16 or 32 (subtypes 14 or 15), then the dialog changes (for both dialogs) are as follows:

1. The "TERM TYPE " prompt now allows the user to specify a descriptor file name in addıtion to the terminal type number as before. A descriptor file describes the characteristics of a terminal as defined by the user. The descriptor file name must be alphanumeric beginning with a letter, but not longer than 8 characters. If the program file is not in the PUB group of the SYS account, then you must fully qualify the program file name. No wild card characters are allowed. The new prompt is:

   ```
   ENTER [TERM TYPE #], [DESCRIPTOR FILENAME]
   ```

   The possible responses are:

   - A RETURN to specify no terminal type number.
   - Enter a terminal type number (as in previous versions of MPE).
   - Enter a descriptor filename and no terminal type number.
   - Enter both a terminal type number and descriptor filename. In this case, the terminal type number is used only in the event that the descriptor file cannot be found.

2. A new portion, beginning with "TERMINAL TYPE CHANGES? ", was added to the dialog; this portion follows the "LIST CS DEVICES? " prompt and precedes all occurrences of the "CLASS CHANGES? " prompt. If you respond "Y " (for yes) to this new prompt, the dialog listed below will appear; if you respond with RETURN, the dialog will take you to "CLASS CHANGES? " at the end of this step. Observe below:

   ```
   TERMINAL TYPE CHANGES? Y
   LIST TERMINAL TYPE FILES? Y       ** Files will be listed here. **
   DELETE TERMINAL TYPE FILES? Y
     FILES?                ** Repeats until you hit RETURN without input. **
   ADD TERMINAL TYPE FILES? Y
     FILE NAME?              ** Repeats until you hit RETURN without input. **
       LOGICAL DEVICE #'S?  ** Repeats until you hit RETURN without input. **
   LIST TERMINAL TYPE FILES?               ** Reply "Y" or "N" as before. **
   CLASS CHANGES?
   ```

In addition to the changes in the INITIAL/SYSDUMP dialog, the configuration display following a " Y " to " LIST I/O DEVICES " is different in some cases. An asterisk (*) will appear in the terminal type column if a descriptor file name is specified for a particular logical device. The terminal type number will also appear if one was specified.

Integrity of IMAGE/3000 Data Bases Greatly Enhanced

Edited by Tim Page

IMAGE/3000 now recovers and restores broken data chains resulting from a system failure, and thus ensures the structural integrity of IMAGE data bases. To enable this feature, Intrinsic Level Recovery (ILR), use the >ENABLE command of the DBUTIL utility.

```
:RUN DBUTIL.PUB.SYS
>ENABLE databasename FOR ILR

>EXIT
:
```

Automatic recovery needs to be enabled for each data base only once. After you perform the steps listed above, IMAGE/3000 creates and maintains an ILR log file in which current intrinsics are stored. If the system fails, the current intrinsic will remain in the log file and be automatically backed out when the data base is next accessed.

The log file name is derived by appending "00" to the root file name. A root file named STORE will have a log file called STORE00. The log file and the root file are automatically stored and restored with the DBSTORE and DBRESTOR utility programs.

Because ILR prevents costly downtime, unloading, and reloading of the data base, HP considers it a valuable feature of IMAGE/3000. A data base enabled for automatic recovery will see a slight (but usually worthwhile) decrease in performance. Refer to "HOW ILR AFFECTS PERFORMANCE" on the following page for details on data base performance.

Automatic recovery is available with IMAGE/3000 version B.04.32, included with the G.00.00 software release.

For more information, refer to the IMAGE/3000 Data Base Management System Reference Manual (32215-90003).

ILR Technical Notes

When automatic recovery is first enabled, DBUTIL creates an ILR log file and sets a flag in the data base root file to indicate that ILR is enabled. DBUTIL also sets the creation data and time in both the root file and ILR file. Data base activities which modify the media record pointers (DBPUT and DBDELETE) are automatically logged in the log file. DBUPDATE activity is not logged because it affects non-key items only.

The Intrinsic Log File Structure

The Intrinsic Log file is a privileged file used by IMAGE only. It contains three parts: the header, the data set users labels, and the data set buffers.

The header, which is 50 words long, maintains the intrinsic-in-progress flag, the intrinsic type (DBPUT or DBDELETE), the data set being modified, etc. The intrinsic-in-progress flag is set when the DBPUT or DBDELETE intrinsic begins execution. The data set user labels require six words per data set. Both the header and the original data set user labels are posted to the ILR file before any data sets or labels are modified by the intrinsic. The data set buffer area starts on the sector boundary after the header and data set user label areas. This area will hold all the buffers that may be modified by one intrinsic.

The ILR Control Block (ILCB)

An extra data segment called the Intrinsic Level Control Block (ILCB) is allocated upon the first DBOPEN of the data base. The ILCB is an intermediate holding area for buffers that will be modified by each DBPUT or DBDELETE.

When a DBPUT or DBDELETE intrinsic executes, data entries to be modified are read into the Data Base Control Block (DBCB) and copied to the ILCB when they need to be posted. Data is then modified in the DBCB. If a failure occurs, and incomplete posting of the modified data results in a broken chain, the data will be recovered from the ILR log file when the data base is next opened. If the modified buffers in the DBCB are posted correctly, the intrinsic-in-progress flag in the ILR file is set to "not-in-progress".

The Recovery Process

When a data base is enabled for ILR, the first data base access by DBOPEN (modes 1 to 8) will examine the ILR file. If the intrinsic-in-progress flag indicates that execution of an intrinsic was interrupted during the last data base access, the ILR log file buffers are posted to the data sets. In that case, the data base is returned to its original state before the interrupted intrinsic began execution.

How ILR Affects Performance

ILR affects IMAGE/3000 performance during DBPUT, DBDELETE, and DBOPEN. The resources taken by ILR are:

- One shared privileged mode file (ILR file).
- One shared extra data segment (ILCB).
- Disc I/O to post the ILCB to the ILR log file.

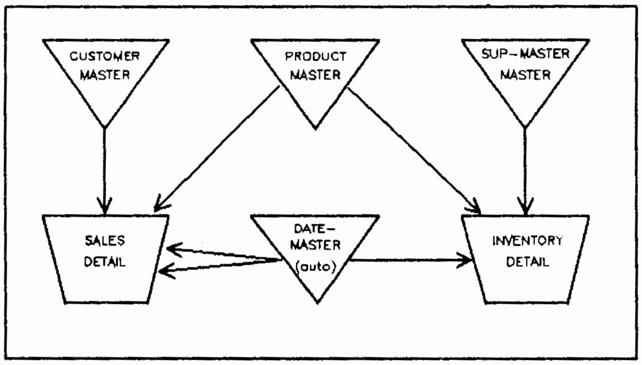
The overhead incurred by ILR depends on the maximum number of buffers that could be modified per data set, which is derived from the data base structure.

For a manual or automatic master data set, the maximum buffers to modify per master set are equal to 4.

For detail data sets, the maximum buffers to modify per detail set are equal to:

```
    (4 * number_of_automatic_paths) + (3 * number_ of_manual_paths) + 1
```

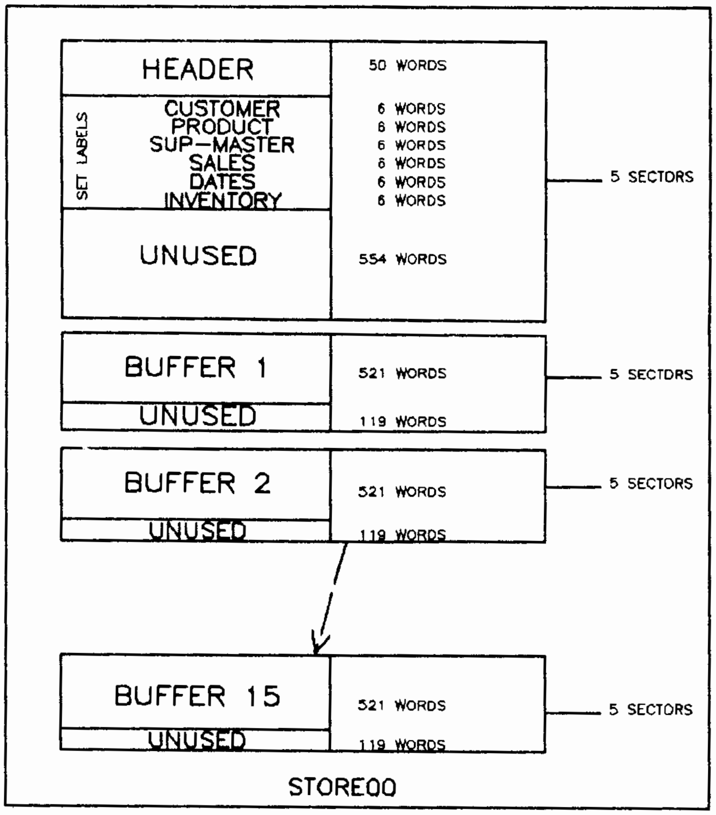
Now, please refer to the STORE data base example in Sections II and III of the IMAGE/3000 Manual (32215-90003), for an elaboration of the example shown on the following page.



In the STORE data base, the maximum number of buffers that could be modified is 15 because the SALES detail has 2 paths from an automatic and two paths from a manual master:

modifiable buffers = (4 * 2) + (3 * 2) + 1 = 15

The maximum size of the buffers is governed by the BLOCKMAX command of DBSCHEMA, and has a default value of 512 words. The actual size is calculated by DBSCHEMA and reported in the table portion of DBSCHEMA's output. For the STORE data base in particular, the calculated buffer length is shown in Figure 3-5 (Section III) of the IMAGE/3000 Manual (32215-90003), is reported as 511 words. Each buffer has a 10-word overhead, for a total of 521 words per buffer, so each log file occupies 5 sectors in the ILR log file (521 words divided by 128 words per sector, rounded up to a whole sector). Note that the header and data set user labels occupy an integral number of ILR log file logical records. Therefore, the log file for the STORE data base would have 15 data buffers, and have a total size of 80 sectors.



An ILR log file holds the buffers for the last intrinsic only, so the log file for the STORE data base needs 15 buffers.

At the beginning of the intrinsics execution, the following occurs:

- One disc I/O records the date, time, data set number, intrinsic-in-progress flag, and data set user labels.

- One or more disc I/Os posts to the buffers, depending on the maximum number of buffers to post and the number of allocated DBCB buffers. For example, if 15 buffers must be posted, but only 8 DBCB buffers are allocated, IMAGE copies the first 8 to the ILR file, posts them to the data base, then copies the remaining 7 from the DBCB to the ILR file and finally posts the buffers. (Use the SET command of the DBUTIL utility to allocate sufficient buffers and improve performance.)

- One disc I/O turns off the intrinsic-in-progress flag in the ILR file.

The minimum number of additional disc I/Os for the DBPUT and DBDELETE intrinsics is therefore three, and the maximum depends on the number of times the DBCB buffer area must be reused.

The maximum number of buffers is used only when all entries that need to be modified reside in different physical disc blocks. This includes migrating secondaries in masters and sorted data chains in the associated detail sets. However, the maximum number of modified buffers possible is seldom reached. IMAGE tries to accommodate synonyms in the same block of master data sets and periodic chained DBUNLOAD and DBLOAD will help position primary key chains of detail data sets within the same physical block.

Some Points about ILR

The following items further describe ILR:

- DBSTORE and DBRESTOR will store and restore the IL file.

- DBINFO has added a new mode, 402, which returns information regarding whether ILR is enabled or disabled, and whether the data base was recovered automatically upon DBOPEN.

- ILR forces the posting of DBCB buffers, so it is not compatible with output-deferred mode. Thus, if the data base is enabled for ILR, a call to DBCONTROL with *mode*=1 (to turn on the output deferred mode) will return a Status of -80. This is a new return status, and it indicates that the attempt to turn on the output deferred option has failed because ILR has been enabled.

  For maximum performance with batch applications, users should do the following:

  1. Turn off ILR in the day's shutdown procedure (or in the job stream that does the shutdown).

  2. Back up the data base.

  3. Run batch applications with DBCONTROL *mode*=1.

- The creator of the ILR file is the creator of the data base, regardless of who enables ILR.

IMAGE/3000 Changes for MPE V/E and Disc Cache

By Doug Griffin

IMAGE/3000 version B.04.02 has been expanded beyond Intrinsic Level Recovery (ILR) to support larger Process Identification Numbers (PINs), larger Extra Data Segment numbers, and disc caching.

To facilitate the MPE V/E expanded tables support of the larger PINs, IMAGE/3000 has expanded the size of the PIN in its internal tables. Because of larger PINs, each lock descriptor in the lock area of the Data Base Control Block will be 2 to 4 words longer. This increase in the size of lock descriptors may cause more occurrences of IMAGE/3000 status 62 :"Lock area is greater than 6144 words" or "Data Base Control Block size is greater than 32600 words". If this occurs, set the "*buffspecs*" lower until the Data Base Control Block size is reduced.

Changes have also been made to support larger Extra Data Segment numbers. IMAGE/3000 now uses a System Data Base Control Block (SDBCB) to hold the Extra Data Segment numbers of all the data bases on the system. The first word of the Base-ID (returned by DBOPEN) will no longer contain the Extra Data Segment number of the Data Base Control Block (DBCB). Instead, the Base-ID will contain the access path number in the first 8 bits and an index into the SDBCB where IMAGE/3000 will locate the Extra Data Segment number of the specific data base. Programs which use the Base-ID for purposes other than IMAGE/3000 intrinsics may need modifications.

When using MPE disc caching's deferred writes, IMAGE/3000's Intrinsic Level Recovery must maintain the chronological order of all writes to synchronize the "before images" with the data base modifications for each intrinsic. To maintain chronological order, IMAGE/3000 uses disc caching's "System Serial Write Queue" whenever Intrinsic Level Recovery is enabled.

When IMAGE/3000 logging is enabled for a data base, DBEND modes 1 and 2 will always wait for the log record to be posted to disc before allowing a program to continue. This will ensure that completed logical transactions will always be recovered during roll forward recovery, and posted to disc before MPE disc caching allows the program to continue. To provide this guarantee, IMAGE/3000 uses the "System Serial Write Queue", which maintains the chronological order of writes. IMAGE/3000 then waits for the last write to complete.

OPT/3000 Disc Caching Enhancements

By Bruna Morosin

The new version A.00.17 of OPT/3000 is offered on MPE V/E. Several enhancements have been made to OPT/3000 as a result of disc caching. As a result, many displays have been modified to illustrate the use of disc caching on various system resources.

In this article, the various modifications to OPT/3000 are summarized. It is recommended that this section be removed from the Special Documentation Package and appended to the current OPT/3000 Reference Manual (32238-90001) until the manual is updated.

Terminology

The following terms will help you to understand the disc caching enhancements made to OPT/3000:

Disc caching: The process of placing the most frequently accessed portions of files and directories into available regions of main memory.

Cached disc domain: A type of segment, analogous to a code or data segment, which may be referenced by MPE disc caching. The maximum attainable size of a cached disc domain is 32K words.

Mapped disc domain: A cached disc domain which is currently being referenced, or is in transit between main memory and disc. The domain also has a mapped entry in the Cache Directory Table.

Unmapped disc domain: A cached disc domain which is neither currently being accessed, nor in transit between the disc and main memory. This domain may be swapped out of memory when space is needed.

Memory Context

Several new displays have been added to the Memory Context to reflect the changes due to disc caching.

MEMORY REPORT DISPLAY. A new field has been added to the ALL MEMORY percentage bar. The " X:Cache Domain " represents the amount of linked memory occupied by cached disc domains.

MEMORY CONTENTS DISPLAY and BANK CONTENTS DISPLAY. Both displays have added the character "X " to their legends to represent the size of each individual cached disc domain. In A.00.12 and previous versions of OPT/3000, upper case letters represented user segments, and lower case letters represented system segments. However, in the case of disc caching, an upper case " X " will denote a mapped disc domain, and a lower case " **x** " will denote an unmapped disc domain.

Global Context

The Global Resource Usage Report contains changes to the MEMORY USAGE percentage bar and the DISC I/O ACTIVITY bar. The MEMORY USAGE percentage bar is now subdivided into five rather than four different states. The added state is " X:Cached Disc Domains ". This field defines the percentage of linked memory occupied by cached disc domains. The DISC I/O ACTIVITY bar has also been modified to reflect disc caching. The " M:Memory Manager " field has been replaced by two new fields: " X:Cached I/O ", and " S:Segment Manager ". The " X " state represents the ratio of disc I/Os completed on the behalf of the disc cache management. The " S " state represents the disc I/O activity completed on behalf of the segment management.

CPU Manager Memory Context

The CPU MANAGER MEMORY CONTEXT has been changed to reflect disc caching.

CPU/MEMORY MANAGEMENT REPORT and CPU USAGE DISPLAY. The CPU STATE percentage bar contains nine rather than eight CPU states. The " M:Memory Allocation " state has been replaced by two new states: " X:Mem Alloc (Cached Management) " and " M:Mem Alloc (Seg) ". The CPU time spent on memory allocation for cache domains is the " X " state, and the CPU time spent on the allocation of segments, is represented by the " M " state.

MEMORY MANAGEMENT ACTIVITY DISPLAY. A new rate bar has been added for disc caching events. The REL CACHE DOMAINS bar represents the number of cache domains released from main memory. The MM I/O READ/WRITE bars have been redefined as SEGMENT READ and SEGMENT WRITE bars representing the number of segment management disc reads/writes performed.

I/O Context and Activity

The I/O Activity Report has new rate bars indicating disc I/O completion rates. The " M: Memory Manager " field has been replaced by two new fields: " X:Cached I/O ", representing the number of disc I/Os completed on behalf of cached disc domains, and " S:Segment Manager I/O ", representing the disc I/Os completed on behalf of segment management.

Process Context

With MPE V/E, the maximum size of the swap table is now 4096. This new size causes a problem with OPT/3000 because if the swap table is configured any larger than 2066, an additional data segment would be required to read in and process the entries. If the configured table size is greater than 2066, instead of using precious resources, OPT/3000 simply uses the portion of the table that can fit in the data segment.

For example, in the USER SUMMARY REPORT (under the columns at the far right, labelled "Working Set Info"), some of the fields may contain "***" instead of numbers, meaning that the swap table entries for these PINs were not available. When viewing individual PINs, you may also see the following message at the bottom of the screen:

```
SOME (OR ALL) SEGMENTS IN THE WORKING SET
ARE NOT SHOWN - Swap Table Overflowed
```

This message does not mean that the swap table overflowed, but that some or all of the entries in the swap table for this process were not available.

OPT/3000 in Batch Mode

Earlier versions of OPT/3000 (before A.00.14) could be run in batch mode but would only give you the GLOBAL STATS SUMMARY REPORT. An enhancement to version A.00.14 allows you to add one of the following parameters onto the RUN command:

```
     PARM = 0      ** Produces the GLOBAL STATS SUMMARY REPORT **
     PARM = 1      ** Produces the DETAILED TABLE INFORMATION REPORT **
     PARM = 2      ** Produces both of the above reports **
```

If you specify an invalid parameter or no parameter, the default gives you the GLOBAL STATS SUMMARY REPORT.

OPT/3000 Log File

New information has been added to the OPT/3000 log file summary records. Programs referencing earlier versions of the log file will not be affected by the added events. All summary records of the log file, except 0, have been modified in the following manner:

WORD **(OCTAL)**	**EVENT**
26-27	Total time (in milliseconds) spent on the user's stack processing caching code since the start of summary report generation.
30-31	The total time (in milliseconds) spent on the Interrupt Control Stack (ICS) processing caching code since summary report generation started.
32-33	The total CPU time spent on disc caching memory allocation.
34-35	Number of disc cache read hits.
36-37	The number of disc cache write hits.
76-77	The number of disc cache domain releases.
	ADDITIONS AS OF RELEASE A.00.10 OF OPT/3000
134-135	All discs with queue length = 0.
136-137	All discs with queue length = 1.
140-141	All discs with queue length = 2.
142-143	All discs with queue length = 3.
144-145	All discs with queue length = 4.

146-147 All discs with queue length = 5.

150-151 All discs with queue length > 5.

ADDITIONS AS OF RELEASE A.00.12 OF OPT/3000

152-153 The number of disc cache reads for all discs.

154-155 The number of disc cache writes for all discs.

EXISTING FIELDS

156-157 The number of memory manager disc reads (Disc 1).

160-161 The number of memory manager disc writes (Disc 1).

162-163 The number of control operations (Disc 1).

164-165 The number of user disc reads (Disc 1).

166-167 The number of user disc writes (Disc 1).

ADDITIONS AS OF RELEASE A.00.10 OF OPT/3000

170-171 Number of times queue length = 0 (Disc 1).

172-173 Number of times queue length = 1 (Disc 1).

174-175 Number of times queue length = 2 (Disc 1).

176-177 Number of times queue length = 3 (Disc 1).

200-201 Number of times queue length = 4 (Disc 1).

202-203 Number of times queue length = 5 (Disc 1).

204-205 Number of times queue length > 5 (Disc 1).

ADDITIONS AS OF RELEASE A.00.12 OF OPT/3000

206-207 The number of disc cache reads (Disc 1).

210-211 The number of disc cache writes (Disc 1).

•
•
•

The following sequence is repeated for each additional disc. The numbers will begin at 212.

EXISTING FIELDS

The number of memory manager disc reads (Disc *n*).

The number of memory manager disc writes (Disc *n*).

The number of control operations (Disc *n*).

The number of user disc reads (Disc *n*).

The number of user disc writes (Disc *n*).

ADDITIONS AS OF RELEASE A.00.10 OF OPT/3000

Number or times queue length = 0 (Disc *n*).

Number of times queue length = 1 (Disc *n*)

Number of times queue length = 2 (Disc *n*).

Number of times queue length = 3 (Disc *n*).

Number of times queue length = 4 (Disc *n*).

Number of times queue length = 5 (Disc *n*).

Number of times queue length > 5 (Disc *n*).

ADDITIONS AS OF RELEASE A.00.12 OF OPT/3000

The number of disc cache reads (Disc *n*).

The number of disc cache writes (Disc *n*).

APS/3000 External Changes

By Phuong Nguyen and Abbas Rafii

With the introduction of MPE V/E with CST expansion microcode, the system and user program spaces of APS/3000 have been enlarged. The table expansions have also increased the maximum number of Process Identification Numbers (PINs) and active program files. These operating system enhancements have necessitated some changes to the external profiles of APS/3000. This article compares the logical address space of a program in MPE IV (or MPE V/P without the new microcode) and MPE V/E, and describes a few changes in the externals of APS/3000.

APS/3000 segment and code profiles are images of program spaces in the system. In MPE IV and MPE V/P (both without CST microcode expansion), the address space of a program consists of 192 system segments and 63 user segments. The system area includes MPE IV (or MPE V/P) segments, subsystem intrinsics and users' Segmented Library (SL) segments. (In this article the term "users' SL segments" refers to the group and public SL segments). The system segments are identified either by their names or by a segment number between 0 and 192 (%300). Similarly, a user segment is identified by a name or a segment number between 193 (%301) and 255 (%377). In this arrangement, the user segments follow the system segments. The APS/3000 profiles reflect this arrangement of the system and user segments.

In MPE V/E with CST microcode expansion, the logical address space of a program consists of two areas, each consisting of up to 255 segments. The first area contains the MPE V/E and certain subsystem segments (physically mapped). The other area contains user segments and users' SL segments (logically mapped). In this arrangement, a given segment number must be further qualified to determine to which area it belongs. In the latter area, the user segments are placed at the lower end starting at segment zero. The SL segments (including dynamic users' SL segments which are loaded at run time) are placed after the last user segment. The boundary between the user and SL segments is, therefore, determined by the maximum number of a program's user segments. The APS/3000 profiles for MPE V/E programs are modified accordingly to reflect the new logical structures.

Log File Compatibility

The log file of each APS/3000 version is only compatible with the version of the program which has created it. Therefore, the old log files are not compatible with the A.01.03 release of APS/3000.

Compatibility with MPE

The A.01.03 release of APS/3000 runs on MPE IV, MPE V/P, and MPE V/E systems. The log file generated on MPE V/E can be analyzed on MPE IV (or MPE V/P) and vice versa, provided that log files are generated by the A.01.03 release of APS/3000.

MPE IV and MPE V/P Programs

The segment and code profiles for programs running under MPE IV or MPE V/P (both without the new CST microcode expansion) are the same as before.

System Segment Definition in MPE V/E Versus MPE IV & MPE V/P

The definition of a system segment is important because it is also used to define indirect cost profiles of APS/3000. Also refer to the description of a program address space given at the beginning of this article.

In MPE V/E with CST microcode expansion, a system segment is defined as being:

- One of the 255 physically mapped MPE V/E and subsystem segments.

- One of the logically mapped users' SL segments (i.e. any segment above the last user segment in the logically mapped area).

In MPE IV (or MPE V/P without the new microcode) the definition of a system segment remains unchanged. That is, a system segment is defined as one of the segments between 0 and 192 (octal 0 and %300).

Definition of Indirect Cost in MPE V/E Versus MPE IV & MPE V/P

Using the definition of a system segment given above for MPE V/E and MPE IV (or MPE V/P), indirect cost of a segment (code, program or PIN) is the fraction of time (samples) a system segment is sampled, and it is found to be called from a user segment (possibly by a sequence of intermediate calls to other system segments). As before, the indirect cost profiles are used to determine the fraction of time system segments are executing on behalf of a user program.

In MPE V/E, the boundary point between the user and system segments in the logically mapped area is not necessarily the same for different programs. Therefore, when more than one program is being sampled, the indirect computation of combined programs cannot be precise. Of course, when a single program is analyzed, the indirect profiles remain valid.

Identifying User and System Segments

In APS/3000 profiles, most of the system segments are identified by their symbolic names. The user segments are also identified by their given names when the PMAP information is available (refer to "SEGMENTER ENHANCEMENTS" in this issue). When both the user segments and system segments are present in an APS/3000 profile, such as the profile for the DA command, the profile starts with physically mapped system segments, followed by user segments, and followed by logically mapped system segments (if any).

Process and Program Profiles

In MPE V/E, the maximum Process Identification Number (PIN) has increased. In order to accomodate the large PINs, the ANALYZER module of APS/3000 uses an internal hashing algorithm. Therefore, the PINs are no longer sequentially listed in process profiles (commands DN and IN in ANALYZER). When the number of distinct sampled PINs is greater than the internal size of the hash table, the samples of the overflow PINs are accumulated in a counter called "Other PINs". For most measurements, the hash table should be large enough to hold all sampled PINs without overflowing into the counter.

Similarly, when the number of distinct sampled programs is larger than the internal hash table size, the overflow program samples are accumulated into a counter labelled Other Programs (commands DF and IF). Again, the user should not see this detail for average measurements.

Other Miscellaneous Changes

Other miscellaneous changes are listed below:

- In SAMPLER, the user can no longer select a subset of segments to be sampled. This feature is deleted because it is rarely used and complicates the dialog with the user in MPE V/E.

- In ANALYZER, for MPE V/E with new microcode, commands IS and TR do not process logically mapped SL segments.

What To Do About JOB OVERLOAD TYPE 6

By Jon Cohen

In the C/D.01.00 release of MPE IV, the system was enhanced to detect two new JOB OVERLOAD conditions: Type 6, UNABLE TO ALLOCATE $STDLIST, and Type 7, UNABLE TO ALLOCATE $STDIN.

Message Explanation

The UCOP (User Controller Process) is the system process that is the father of all Command Interpreter (CI) processes (jobs and sessions). The UCOP's basic strategy is to try to obtain all resources needed by a CI process before launching the job or session. If it is unable to acquire all needed resources, a job is placed back in the WAIT state, and a session would probably receive a " CAN'T INITIATE NEW SESSIONS " message. In either case, the " JOB OVFRLOAD " message is issued on the system console.

Part of the initialization that UCOP performs is the pre-allocation of the $STDIN and $STDLIST devices. If either of these fail, the new " JOB OVERLOAD " message is issued. (For sessions, a message can not be printed on the terminal, since it could not be allocated, and the job will return to the WAIT queue.)

In practice, the " JOB OVERLOAD TYPE 6 " message has been seen most often. Since these often occur for jobs, the jobs are placed back in the WAIT queue. When the user tries to log on again, the " JOB OVERLOAD " state is detected, and another message appears on the system console. These messages will continue to appear on the system console until the error condition is corrected either by system operator intervention or by the system itself.

Troubleshooting

The message " JOB OVERLOAD TYPE 6 " is issued each time a job can't allocate $STDLIST; therefore, the message on the console will repeat constantly. Usually this condition arises when the spooling queues shut due to a spooling error. Here are some steps to detect and solve the problem:

1. Enter: :<u>SHOWJOB JOB=@J</u>

   If there are jobs in the WAIT state and the number of jobs in EXEC state is less than the job limit, this is a good indication that the spool queues have shut down.

2. Each job in the WAIT state will have a JLIST device listed. Use the :SHOWDEV command to see if the device is still spooled. For example, if the JLIST device is 6, enter:

   :<u>SHOWDEV 6</u>

   The output will reflect whether the device is spooled or shut down.

```
6    SPOOLED    SPOOLER OUT   ** Device spooled; JOB OVERLOAD state is
                                 elsewhere. **

6    UNAVAIL    SPOOLER OUT   ** Device spooler queues are down. **
```

3. Reopening a shut spool queue will allow previously failed jobs to log on. This assumes that the spool queues stay open. Occasionally spool queues will shut down due to lack of system disc space. While you may be able to open the queues again, unless more disc space becomes available, the queues will shut down again. In order to free disc space, it is often sufficient to use the SPOOK utility to store low priority spoolfiles onto tape for printing at a later time.

   Another reason that might cause the spooler queues to shut would be that there is no more spoolfile space available on the system. If this is the case, you should first try to print or purge any large spoolfiles. Next, you would want to `:RUN FREE5.PUB.SYS` to determine if the available disc space is fragmented. If there is not enough disc space, you would then want to `=SHUTDOWN` and `COOLSTART` with the `"RECOVER LOST DISC SPACE"` option.

4. Reopening the spool queues is now a simple process on MPE V/E (G.00.00). Use the `:OPENQ` command with the *ldev* or *devclass* parameters. (For more information on this command, refer to "NEW :OPENQ AND :SHUTQ COMMANDS" in Section II.)

   On pre-MPE V/E versions of the operating system, reopening the spool queues is not simple, since it is complicated by the extra efforts needed to prevent users from accidentally acquiring the device as unspooled ("hot"). Again, using logical device 6 as an example, enter the following commands to prevent this:

   ```
   :OUTFENCE 14;LDEV=6
   :JOBFENCE 14
   ```

   These commands will prevent anyone from logging on while you are reopening the spool queues. Next, you will have to stop spooling to that device; use the following command:

   ```
   :STOPSPOOL 6
   ```

   Make sure that you are using the number of the logical device and not the device class name. It may be necessary to perform some `:ABORTIO 6` commands before the spooler will stop. When done, the result of a `:SHOWDEV 6` should show:

   ```
   6    AVAIL
   ```

   Now that the device is available, you can restart spooling to it with the following command:

   ```
   :STARTSPOOL 6
   ```

   Once that is done, reset the OUTFENCE and the JOBFENCE. For example:

   ```
   :OUTFENCE 5
   :JOBFENCE 2
   ```

   This allows jobs to log on and spoolfiles to use the device.

MPE Stack Size Requirements

By Gary O'Neall

The stack size requirements for file system intrinsics have increased from C/D.01.00 to G.00.00. The increased system table sizes necessitated this increase in stack usage. This may cause some programs currently running at or near their stack limit to abort with a stack overflow. It may be necessary to increase the initial stack size and/or *maxdata* values or try running programs with the ;NOCB option in order to successfully run these programs under G.00.00. It may also be desirable to reconfigure either the maximum or standard stack size values to be larger.

The following table compares C/D.01.00 to G.00.00 stack requirements for file system intrinsics:

	New disc file;save		Old disc file		Dev = LP no ENV		Dev=EPOC ENV	
FILE SYSTEM INTRINSICS	OLD	NEW	OLD	NEW	OLD	NEW	OLD	NEW
FOPEN	1054	1135	1054	1135	1089	1294	2271	2379
PRINTFILEINFO	578	665	578	665	442	484	441	484
FFILEINFO	387	397	387	397	406	421	406	421
FGETINFO	224	229	224	229	243	253	243	253
FWRITE	344	519	344	519	226	221	226	221
FREADDIR	229	249	229	249	-	-	-	-
FWRITEDIR	404	569	350	569	-	-	-	-
FCONTROL	197	211	197	211	-	-	-	-
FREAD	384	541	348	541	-	-	-	-
FCLOSE	925	929	925	929	925	929	925	929

NOTE

All values are in words. The stack size required for an FOPEN on a private volume is now 1296 words.

SEGMENTER Enhancements

By Jessey Hsu

Editor's Note: The information in this article (which was published in the Communicator, Issue 30) was finalized after the new (November 1982) edition of the MPE Segmenter Reference Manual (30000-90011) was printed. On page 2-74 of Communicator Issue 30, a table about the ;FPMAP option of the SEGMENTER contained incorrect information. The correct table, with changes underlined, is included in this article (see Table 1).

This information will be incorporated in the next update to the manual.

Most experienced users of MPE are familiar with the PMAP option of the Segmenter -PREPARE and -ADDSL commands and the system :PREP command. These cause the Segmenter to produce a listing of the locations of procedures and their entry points, plus external references, in all segments prepared.

The PMAP information most useful to the user is that which identifies the locations of procedures and their entry points within a given code segment. We have now added a mechanism to store an internal copy of most of the information contained in the PMAP as part of a program or SL file; this is called the FPMAP information. External references to procedures in other segments are not saved as part of the program and SL file FPMAP information. A new command and new user-callable procedures are provided to access the FPMAP information.

This information is especially helpful for users of APS/3000 and HPToolset. The inclusion of the FPMAP information as part of a program or SL file provides a mapping of segment-relative addresses back to the source-level procedure names and entry points. Application Program SAMPLER (APS/3000), for instance, uses this information to provide detailed execution profiles of any monitored program, in terms of procedure names and code within each procedure. This information can be used to quickly determine the CPU bottleneck of almost any program on a HP/3000 system. When the System FPMAP flag is set, the FPMAP data is automatically included in each program file at PREP time. Thus, the procedure execution profile of any active program in the installation can be readily obtained by the installation manager or development engineers.

Performance Considerations

The program file size increases about 3 to 10 percent if the FPMAP information is included in the file. CPU time for preparing a program increases about 5 to 20 percent when a program is prepared with FPMAP.

Controlling the Inclusion of FPMAP Information

Several new features have been added to the Segmenter which allow System Managers and users to control the FPMAP inclusion in program and SL files. The inclusion of the FPMAP information depends on the conditions of the System-wide FPMAP option flag and the Job/Session-wide FPMAP option flag, and the new FPMAP and NOFPMAP parameters of the :PREPARE, -PREPARE, and -ADDSL commands. Table 1 illustrates the interaction.

Notice that the command parameters override the Job/Session FPMAP flag, and the Job/Session FPMAP flag overrides the System FPMAP flag, unless the System FPMAP flag is set to unconditional.

The System FPMAP flag can be set to conditional or unconditional. If it is set to unconditional, all users on the system will have FPMAP information included in their programs and SL files when they prepare programs or add segments to SL files, regardless of the Job/Session FPMAP flag and parameters specified. If the System FPMAP flag is conditional, then it can be overridden by lower levels of FPMAP option control. For example, when either the System conditional FPMAP or the Job/Session FPMAP is on, the user can enter the NOFPMAP parameter. This will result in a program without FPMAP informat. The System FPMAP flag is initialized to OFF when a system is started or loaded. When a job or a session is logged on, the Job/Session FPMAP flag is initialized to the System FPMAP flag value.

Table 1. Interactions in FPMAP Inclusion

System FPMAP Flag	**Job/Session FPMAP Flag**	**FPMAP/NOFPMAP Parameter**	**FPMAP Included?**
Uncond. On	On or Off	Either	Yes
Cond. On	On	Neither	Yes
		FPMAP	Yes
		NOFPMAP	No
	Off	Neither	No
		FPMAP	Yes
		NOFPMAP	No
Off	On	Neither	Yes
		FPMAP	Yes
		NOFPMAP	No
	Off	Neither	No
		FPMAP	Yes
		NOFPMAP	No

For users of the HPToolset subsystem, the FPMAP will always be included if symbolic debugging is requested, regardless of the flags specified. Only the NOSYM parameter can suppress the FPMAP information in this instance. (Refer to Communicator Issue Number 29, "NEW PREP COMMAND PARAMETERS".)

The new facilities for FPMAP inclusion are:

- A new optional parameter pair has been added to the -PREPARE, :PREPARE, and -ADDSL commands; the default conditions can be determined from Table 1. The syntax is:

  ```
  -PREPARE progname [;map_opt]
  -PREPARE progname [;map_opt]
  -ADDSL segname [;map_opt]
  ```

  where:

  ```
  map_opt is defined as  {FPMAP  }
                         {NOFFMAP}
  ```

- The -SETFPMAP command has been added to the Segmenter to set the system or job/session FPMAP flag. The syntax is:

```
                                {CONDITION  }
            {SYSTEM} [;                      ] }            {ON }
-SETFPMAP  [                    {UNCONDITION}       ][;           ]
            {SESSION                           }            {OFF}
```

where:

SYSTEM/SESSION — Specifies which flag is to be set/reset. Default is SESSION. Requires SM capability to set/reset System FPMAP flag.

CONDITION/UNCONDITION — Specifies that the System FPMAP flag is set to conditional or unconditional. Valid only when SYSTEM is specified. Default is CONDITION.

ON/OFF — Set/Reset flag. Default is ON.

- The -SHOW command has been added to the Segmenter to display the conditions of the System and Job/Session FPMAP flags. In addition, it lists files currently opened by the Segmenter. The syntax is:

```
-SHOW
```

An example output is:

```
-SHOW

   USL FILE        :  MYUSL.MYGROUP.MYACCT
   AUX USL FILE    :  NONE
   SL FILE         :  MYSL.MYGROUP.MYACCT
   RL FILE         :  NONE
   SYSTEM FPMAP    :  ON (CONDITION)
   SESSION FPMAP   :  OFF
```

Accessing the FPMAP Information Interactively

FPMAP information may be accessed interactively through the new -LISTPMAP Segmenter command. The command's syntax is:

```
                      {segname       }
-LISTPMAP progname [;{procedurename}]
```

where:

progname Is the name of the program whose PMAP is to be listed.

segname Is the name of the segment whose PMAP is to be listed.

procedurename Is the name of the procedure whose PMAP is to be listed.

The default is to list all segments in the specified program file.

An example output follows:

```
-LISTPMAP MYPROG

PROGRAM FILE MYPROG.MYGROUP.MYACCT

MAIN            0       4056

NAME           TYPE     CODE     ENTRY     LENGTH
PROC1           P          0        10         45
PROC11         SP                   21
PROC2           P         45        45       3000
PROC3           P       3045      3050       1000
```

Obtaining FPMAP Information Programmatically

A new set of user-callable procedures has been added to allow programmatic access to FPMAP information. They must be declared explicitly within the calling program. Declarations appear in the description of each procedure.

In order to provide a unified means of returning FPMAP data, each of these procedures makes use of a special data structure called an external FPMAP record, whose format is given in Figure 1. (All undefined fields will contain zeros or blanks as appropriate.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Field Defined
	Type (0=Segment, 1=Procedure, 2=Sec. Entry)	0, 1, 2
1 8	Segment Name	0, 1, 2
9 16	Procedure Name	1, 2
17 24	Secondary Entry Point Name	2
25	Logical Segment Number	0, 1, 2
26	Segment Length (Including STT)	0, 1, 2
27	STT Length	0, 1, 2
28	Procedure Start Address	1, 2
29	Procedure Length	1, 2
30	Primary Entry Point Address	1, 2
31	Secondary Entry Point Address	2
32	(Not Used)	
33	HPToolset Procedure ID	1
34 35	HPToolset Link	1

Figure 1. External FPMAP Record Format

FINDPMAPNAME. This procedure allows programmatic access to FPMAP information.

Declarations. The FindPmapName declarations are:

```
FindPmapName (ProgFnum, SegNamesEntName,XpmapRec,XpmapRecLen,Status);
    VALUE ProgFnum, XpmapRecLen;
    INTEGER ProgFnum;
    BYTE ARRAY SegName;
    BYTE array EntName;
    INTEGER ARRAY XpmapRec;
    INTEGER XpmapRecLen, Status;

    OPTION VARIABLE;
```

Description. This procedure searches the FPMAP in a program or SL file for a segment and/or entry point name. If found, an appropriate external FPMAP record will be returned. Otherwise, the status code will indicate why the search failed.

One or both of the parameters *SegName* and *EntName* must be included in the actual parameter list; the action taken according to their various combinations is as follows:

SegName only - This calling sequence results in a search for a segment name only. If found, a segment (type 0) external FPMAP record will be returned.

EntName only - This calling sequence results in a search for a procedure or secondary entry point name. In program files, all procedures and secondary entry points qualify for the search. In SL files, only those procedures and secondary entry points appearing in the file directory will be candidates for a match. This implies that hidden procedures and secondary entry points will not be found. Only a procedure (type 1) or entry point (type 2) external FPMAP record will be returned.

Both *SegName* and *EntName* - This calling sequence results in a search for a procedure or secondary entry point name in a specified segment. Hidden procedures and secondary entry points will be included in the search since the segment name qualifies such names for uniqueness.

SegName and *EntName* - These may be excluded from a procedure call either by physical omission in the parameter list, or by the presence of a blank as the first character in their byte arrays. All other parameters in the calling sequence must be present.

Input Variables. FindPmapName input variables are:

ProgFnum - This is the number of the program or SL file, as returned from the FOPEN intrinsic, whose FPMAP is to be searched. The caller is responsible for opening the file with MULTIRECORD, NOBUF, and READ access.

SegName - This is the name of the segment to be found. It should be terminated by a blank, but in no case will more than the first 15 characters be accessed for the search. Comparisons will be made with all characters upshifted, but the string passed in *SegName* will be returned unchanged. This is an optional parameter, and interacts with *EntName*. Refer to the discussion above for details.

EntName - This is the name of the procedure or secondary entry point to be found. The format restrictions given for *SegName* also apply to this variable. This is an optional parameter, and interacts with *SegName*. Refer to the discussion above for details.

XpmapRecLen - This is the number of words of the external FPMAP record the caller wants returned in *XpmapRec*. Should the caller request more words than are currently defined for an external FPMAP record, zeroes will be returned in the extra words.

Output Variables. *FindPmapName* output variables are listed below:

XpmapRec - This is the external FPMAP record corresponding to the entry point name passed in "EntName". The format of this record has been previously defined in this document. Any of the external FPMAP record types may be returned.

Status - This is a status word containing information about the success or failure of the procedure call as follows:

0 - No errors.

1 - The entry point name was not found.

3 - The segment name was not found.

9 - An invalid parameter combination was passed.

10 - The file *ProgFnum* did not contain FPMAP data.

11 - The file *ProgFnum* was not a program or SL file.

12 - A file system I/O error occurred on the file *ProgFnum*.

Condition Code. Following are condition codes used with FindPmapName :

CCE - The symbolic name was found (Status = 0).

CCG - The symbolic name was not found (Status = 1 or 3).

CCL - Unusual error (Status = 9 through 12).

FINDPMAPADDR. This procedure allows programmatic access to FPMAP information.

Declarations. The declarations for FindPmapAddr are:

```
FindPmapAddr(ProgFnum, SegNum, Address,XpmapRec, SpmapRecLen, Status);
     VALUE ProgFnum, SegNum, Address, XpmapRecLen;
     INTEGER ProgFnum, SegNum, Address;
     INTEGER ARRAY XpmapRec;
     INTEGER XpmapRecLen, Status;
```

Description. This procedure searches the FPMAP in a program or SL file for a specific address in a particular segment. An external FPMAP record corresponding to the first entry point at or preceding *Address* is returned.

Input Variables. FindPmapAddr input variables are:

ProgFnum - This is the number of the program or SL file, as returned from the FOPEN intrinsic, whose FPMAP is to be searched. The caller is responsible for opening the file with MULTIRECORD NOBUF READ access.

SegNum - This is the number of the logical segment in which the address is to be found.

Address - This is the PB-relative address in segment *SegNum* with which the external FPMAP record to be returned is associated.

XpmapRecLen - This is the number of words of the external FPMAP record the caller wants returned in *XpmapRec*. Should the caller request more words than are currently defined for an external FPMAP record, zeroes will be returned in the extra words.

Output Variables. FindPmapAddr output variables are:

XpmapRec - This is the external FPMAP record corresponding to the first entry point at or preceding *Address*. The format of this record has been previously defined in this document. It may be either a procedure (type 1) or secondary entry point (type 2) record, depending on *Address* and whether or not secondary entry points exist. If the located entry point has more than one name, the first one encountered in the FPMAP will be returned.

Status - This is a status word containing information about the success or failure of the procedure call as follows:

0 - The address was valid and an external FPMAP record was returned.
2 - The address was out of range.
3 - Logical segment *SegNum* didn't exist.
10 - The file *ProgFnum* did not contain FPMAP data.
11 - The file *ProgFnum* was not a program or SL file.
12 - A file system I/O error occurred on file *ProgFnum*.

Condition Code. Following are the condition codes for FindPmapAddr.

CCE - The address was valid (Status = 0).

CCG - The address or segment number was invalid (Status = 2 or 3).

CCL - File error (Status = 10 through 12).

DUMPPMAP. This procedure allows programmatic access to FPMAP information:

Declarations. The declarations for DumpPmap are:

```
DumpPmap (ProgFnum,PmapFnum,XpmapRecLen,RecCount,SegNum,Status);
     VALUE ProgFnum, PmapFnum, XpmapRecLen, SegNum;
     INTEGER progFnum,PmapFnum,XpmapRecLen,RecCount, Segnum, Status;
```

Description. This procedure fills a file with external FPMAP records representing either all of the FPMAP data stored in a program or SL file, or only that for a specified segment. One external record will be created for, and in the same order as, each internal FPMAP record encountered.

Note that if multiple names exist for the same secondary entry point address, one FPMAP record will be created for each of the names. DumpPmap is the only procedure which provides access to such multiply defined names.

Input Variables. DumpPmap input variables are:

ProgFnum - This is the number of the program or SL file, as returned from the FOPEN intrinsic, whose FPMAP is to be dumped. The caller is responsible for opening the file with MULTIRECORD, READ-ONLY access.

PmapFnum - This is the number of the file, as returned from the FOPEN intrinsic, which is to contain the external FPMAP records. The caller is responsible for opening the file with write access and fixed-length records of size at least *XpmapRecLen* words.

XpmapRecLen - This is the number of words in each external FPMAP record to be written to file *PmapFnum*. Should the caller request more words than are currently defined for an external FPMAP record, the extra words in each record written will be set to zero.

SegNum - This is the number of the segment whose FPMAP data are to be dumped. A value of -1 causes all FPMAP records for all segments in the file to be dumped.

Output Variables. DumpPmap output variables are:

RecCount - This is the number of external FPMAP records which were, or could have been, generated from the FPMAP data found in the program or SL file. Assuming the caller's *PmapFnum* file was large enough, this is also the number of records written to that file.

Status - This is a status word containing information about the success or failure of the procedure call as follows:

0 - The external FPMAP file was built successfully.
4 - The external FPMAP file was not large enough.
10 - The file *ProgFnum* did not contain FPMAP data.
11 - The file *ProgFnum* was not a program or SL file.
12 - A file system I/O error occurred on the file *ProgFnum*.
13 - A file system I/O error occurred on file *PmapFnum*.

Condition Code. The condition codes for DumpPmap follows:

CCE - The FPMAP file was built successfully (Status = 0).

CCG - The FPMAP file was not large enough (Status = 4).

CCL - File error (Status = 10 through 13).

DATA COMM ENHANCEMENTS

SECTION VI

MPE's IML Commands Are Now IMF Commands

By Jan Helmbolt

When the IML subsystem originally was released as part of the MPE operating system, the MPE commands associated with the operation of the subsystem were :IML , :IMLMGR , and :IMLCONTROL . When the subsystem changed names from IML to IMF, the commands associated with it also changed their names to :IMF , :IMFMGR , and IMFCONTROL . As part of Hewlett-Packard's commitment to backwards compatability, MPE has continued to support the IML commands as part of the operating system, even though the commands have been documented as " IMF... ".

To eliminate any further user confusion, the commands :IML , :IMLMGR , and :IMLCONTROL have been removed from the MPE operating system and have been permanently replaced with :IMF , :IMFMGR , and :IMFCONTROL respectively. Therefore, users wishing to use :ALLOW with the operator command :IMLCONTROL must now enter:

```
:ALLOW user;COMMANDS=IMFCONTROL
```

DS With Enhanced X.25 Capability

By Carol Hibbard

New Product Structure

DS and X.25 for the HP 3000 became separate products on the recently released product tape. The new product structure takes the capability of the HP 32190 DS product and divides it into the HP 32189 DS and HP 32191 X.25 products. The new HP 32189 DS product provides the exact same capabilities provided by DS/3000: virtual terminal, remote file and database access, network file transfer, and program-to-program communication while using the Bisync Protocol. The HP 32191 X.25 product supports terminal-to-system communications over X.25 Packet Switched Networks and provides an X.25 interface for HP 32189 DS to allow system-to-system communications using the X.25 protocols. Therefore, both HP 32189 DS and HP 32191 X.25 must be installed on your system to allow system-to-system communication across X.25 Packet Switched Networks or full-duplex point-to-point links using the X.25 protocols.

Support Contracts

The new arrangement decreases the cost for those who have had to purchase DS in order to connect terminals to the HP 3000 over an X.25 Packet Switching Network. Also, the separation of these products offers a reduction in monthly support costs for those not using the X.25 capability of DS/3000. You will have the choice of continuing your HP 32190T/S/V support contracts or writing new software support service contracts for covering the HP 32189 and/or HP 32191 product(s). If you are covered by the new Software Support Services program, Software Materials Subscription would be amended and an appropriate Datacomm Category Support would be selected.

If you are not using the X.25 capability of HP 32190, you can save money by amending your support contract to delete 32190T and add 32189T. Similarly, if you are only using X.25 for terminal-to-system communications, then the cancellation of 32190T and the purchase of 32191T would also save you money. The total cost of 32189T and 32191T is the same as for 32190T. The price for new Software Support Services depends on software categories represented on a system; for more information on support contracts, contact your local HP Sales Office.

Enhanced X.25 Capability

X.25 for the HP 3000 computer (32191) has been enhanced to support block-mode applications written in HP VPLUS and the new HP 2334A X.25 Cluster Controller. Previously, X.25 supported only character-mode applications. With the block-mode enhancement, remote sites with terminals may run VPLUS applications, enter blocks of data with a single keystroke, and transmit it to an HP 3000 computer over an X.25 Packet Switched Network (PSN). This block-mode capability is ideal for applications such as remote sales offices that need to place orders at a central distribution site. For more information on this new block-mode capability, refer to the X.25 for the HP 3000 Reference Manual (32191-90001).

With the support of the new HP 2334A X.25 Cluster Controller, you now have two possible solutions for system-to-terminal X.25 communications. If there are only one to three terminals at the remote site, you can continue to dial up the PSN PAD for connection to HP 3000 computers. If you have more than three terminals and/or printers at this remote site, then the use of the HP 2334A X.25 Cluster Controller, which connects to the X.25 PSN with just one leased line, offers a more cost-effective alternative. The HP 2334A X.25 Cluster Controller can support a maximum of 16 terminals or nonspooled printers (HP 2601, 2602, or 2631B). For more information contact your local HP Sales Representative.

New Manuals

The unbundling of X.25 from DS means that the DS manual has been unbundled as well. Users of X.25 should use the X.25 for the HP 3000 Reference Manual (32191-90001). Users of DS will receive two manuals to replace the current DS manual when they receive their new, unbundled version of DS. These manuals are the DS User/Programmer Reference Manual for the HP 3000 (32189-90001), and the DS Network Administrator Manual for the HP 3000 (32189-90002). The X.25 manual is intended for those customers who use only X.25, while the two DS manuals are for those customers who use DS alone or in conjunction with X.25 for their applications. In addition, a new edition of the DS HP 3000 to HP 1000 Reference Manual (32189-90005) will be appearing at the same time.

MTS Allows Use of New Devices

By Mary Ann Adams

HP 2333A Cluster Controller

The HP 2333A cluster controller on a multipoint network allows the attachment of up to 16 point-point devices to communicate with the host computer, either via the HP Data Link, or via modems and leased lines. The attached devices operate much as they would if they were directly connected to an HP 3000 terminal interface (such as the ATP, ADCC or ATC).

It is possible to attach up to 32 HP 3081A terminals to the HP 2333A cluster controller. Changes have been made in MPCONFIG and MPLINE syntax in order to be able to configure 32 terminals to the cluster controller. The reason these changes were necessary is that the cluster controller identifies the additional ports with lower-case letters. The first 16 ports are identified with upper-case letters. MPCONFIG and the MPLINE processor do not differentiate between upper- and lower-case characters. Therefore, with the new syntax, the ports are identified by MTS as numbers 1 through 32 rather than "A" through "P" and "a" through "p".

Enhanced MTS Capability

There have been a number of other enhancements to the MTS product. MTS allows users to connect serial printers to a multipoint line for low-speed light-duty applications. The printers can be spooled or run hot. They may be connected to the multipoint line in one of three ways: directly to a MTS link; to an HP 2333A cluster controller on a MTS link; or to an HP 2624B terminal in terminal by-pass mode on a MTS link. The printers must have a full duplex RS-232-C interface. When the printers are connected to the HP 2624B terminal, they are limited to a maximum speed of 2400 bps, and when they are connected to the HP 2333A, they have a maximum speed of 9600 bps. A maximum of four serial printers can be attached to each multipoint line. For each multipoint line that has a serial printer, the user must create an MPCONFIG file. The line can now be configured to run as either odd parity (same as older versions) or as no parity. When configured as no parity, 8-bit codes can be sent to the HP terminals that refuse to recognize 8-bit data when configured for anything but parity=none. The user can now specify in the MPCONFIG generated configuration file that a timeout is to be used when waiting for a response to a poll or selection of a device. The default timeout remains three seconds, as in the past. This parameter will allow you to tailor the poll and select timeout to the actual configuration in use.

The chart on the following page shows how the various printers are supported with MTS.

Printers Supported with MTS

Printer Model No.	Type of Printer Interface	Method of Connection	Level of Support	Max. Speed Supported
HP 2601A HP 2602A	Point-to-Point	HP 2333A HP 2624B	Auxiliary Printer (termtype 18)	1200 bps
HP 2608S	Multipoint Daisy Chain or Data Link	Directly to Multipoint Line	System Printer (using Ciper protocol)	19,200 bps
HP 2631B	Point-to-Point	HP 2333A HP 2624B	Remote Spooled Printer (termtype 22)	2400 bps
HP 2563A	Point-to-Point	HP 2333A only	Remote Spooled Printer (termtype 22)	9600 bps
	Multipoint Daisy Chain or Data Link	Directly to Multipoint Line	System Printer (using Ciper protocol)	19,200 bps
HP 2932A	Point-to-Point	HP 2333A or HP 2624B	Remote Spooled Printer (termtype 22)	2400 bps
HP 2933A HP 2934A	Point-to Point	HP 2333A or HP 2624B	Remote Spooled Printer (termtype 22)	2400 bps
	Multipoint Daisy Chain or Data Link	Directly to Multipoint Line	Auxiliary Printer (termtype 18)	9600 bps

Workstation Configurator

By Jim Talley

The Workstation Configurator, WSC, is a new product which allows an HP 3000 user to manage and configure system I/O ports for connection to different types of workstations. With the WSC, users of HP 3000 systems with Advanced Terminal Processors (ATPs) and/or Advanced Data Communications Channels (ADCCs) can define and save characteristics pertaining to a given workstation. WSC supports user-definable terminal type files that directly control the table-driven ATP software driver. A user may set and programmatically change such workstation characteristics as parity, flow-control handshakes, special function characters, TTY delays, and read trigger characters. In doing so, a user can effectively manage his I/O ports to take advantage of specific workstation capabilities and connect otherwise incompatible devices to the HP 3000. (Refer to "INITIAL/SYSDUMP DIALOG CHANGES WITH THIS RELEASE" for an explanation of the changes made to these dialogs to accomodate the Workstation Configurator software.)

Features

The Workstation Configurator has the following features:

- A Workstation Configurator utility routine with an interactive, menu-driven interface which creates, modifies, and manipulates terminal type files.
- The user can specify workstation flow-control protocols: Enquiry/Acknowledge or Delay handshakes controlled by the software driver, and the XON/XOFF mechanism controlled by the attached device.
- The line or page block-mode terminals are supported by allowing the user to define the block-mode alert and trigger characters.
- The user can specify what special characters to use for system attention, backspace, cancel-line, end-of-record, and subsystem-break.
- The user can define control settings such as echo, line-feed, backspace response, and parity.
- The user can define characters to be stripped and ignored by the I/O driver.
- Printer control information such as initialization strings and Vertical Format Controls can be defined.

Functional Description

In the current ATP and ADCC products, all the characteristics pertaining to a given workstation are predefined by the system by assigning a terminal type number that identifies the I/O needs of the specific device (see Terminal Type Table in ATP data sheet). The predefined terminal types restrict a user from configuring the ATP or ADCC I/O drivers to take maximum advantage of a workstation's functions and capabilities. The Workstation Configurator software removes this fixed terminal type restriction. It allows a user to define his own terminal types and save the information in disc files for future use.

WSC provides an interface for the user to configure a particular terminal type. An interactive, menu-driven session allows a user to create or modify terminal types. The user-defined terminal type can then be saved as a disc file and assigned to a particular I/O port at system configuration time as a default value. It can also be used at logon to override any existing default terminal type.

Three flow-control handshakes are available to the user:

HANDSHAKE	MEANING
Enquiry/Acknowledge	The protocol that allows the user to define the Enquiry/Acknowledgement (ENQ/ACK) characters, block size, and time-out option.
Delay	The protocol that allows the user to specify the delay value for carriage-return, line-feed, and form-feed.
XON/XOFF	The protocol that defines the time-out value.

WSC allows a user to specify whether block-mode support is needed, and what characters are used for alerting and triggering a block read. The I/O driver-specific characters, such as MPE console-attention, backspace, cancel-line, end-of-record, and subsystem-break can all be redefined using WSC. A set of characters which has no special function, other than to be removed from the input data, can also be defined. Other definable characteristics in WSC include control settings such as echo, backspace response, parity, initialization string, and Vertical Format Control for the printer.

Product Requirements

The Workstation Configurator requires the following products to run:

- The G.00.00 (or later) version of the MPE V operating system.
- Workstation Configurator Software (30239A/M/R).
- Advanced Terminal Processor Hardware (30144A, 30145A, and 30155A) or Asynchronous Data Communications Controller (30018A, 30019A).

The subsystems Supported are: HPSLATE, TDP, DSG, HPDRAW, HPEASYCHART, HPMAIL, Rapid Products, and VPLUS. Workstation Configurator will not be supported across DS/3000 lines.

Support Policy

When using the WSC, one can configure an HP 3000 to connect many different device types that have many varied characteristics. However, only HP workstations described in the ATP data sheet are supported by HP. Other devices may be connected to an HP 3000, but it will be the customer's responsibility to ensure device compatibility with the ATP and ADCC hardware and the WSC software. WSC support guarantees that the software will perform all the functions described in the data sheet. Support of any non-HP devices on the HP 3000 using WSC remains the responsibility of the customer. If problems are encountered in utilizing WSC, HP's Field Support will, at the customer's option, verify the operation of WSC. Any effort expended against problems related to facilities not a part of WSC will be considered HP consulting support, and thus billable to the customer at normal HP consulting rates.

Installation Policy

HP provides installation as part of the license for the 30239A product of WSC. ("Non-A" products do not include installation as part of the product.)

Prior to the WSC installation, customers must perform an HP 3000 system backup, including @.PUB.SYS , and make the HP 3000 system available for WSC installation and test.

During the WSC installation, HP will perform a System Update to add the product software modules to the system and verify that the WSC utility runs properly. Note that at this point, installation of the WSC product is considered complete.

Ordering Information

The following software and support products are available to order for the Workstation Configurator:

- 30239A WSC (license to use).
- 30239M WSC (additional site license).
- 30239R WSC (right to copy and sublicense one copy).
- 30239T (comprehensive software support).
- 30239V (additional site support).
- 30239S (software subscription service).
- 30239Q (manual update service).

Documentation

For further technical information, please consult The Workstation Configurator Reference Manual (30239-90001) or The Point-to-Point Workstation I/O Reference Manual (30000-90250).

MPE V/E ADCC Terminal Driver Changes

By Jim Talley

New ADCC software driver code will be released with MPE V/E. The new software code is represented by two new driver names: HIOTERM2 (handles device type 16) and HIOASLP2 (handles device type 32). The conversion program (PCONVERT.PUB.SYS) changes the driver names in the I/O configuration when updating to MPE V/E or when regressing to an earlier MIT. The old driver (HIOTERM0) will not work with MPE V/E. The new ADCC code has become an integral part of the ATP driver modules; the new module names and numbers are as follows:

Module Name	Module #
ADCCINIT	60
ATPINIT	61
LPMON	67
TERMMON	68
IMANAGER	89
IHANDLER	90
TERMUTIL	80
ADCCDRIVER	95
ATPDRIVR	94

In addition, ATPDSM has been changed and can now be used with an ADCC; therefore, we changed the name of the on-line diagnostic to TERMDSM.

Advantages

When compared to the existing driver, the new code offers the following advantages:

- Up to 12000 words of Bank 0 have been freed for other uses; that is, all the TBUFs and most of the DIT can now be found in an extra data segment very similar to the ATP XDS.

- The size of terminal buffers (TBUFs) have increased from 60 bytes to 134 data bytes, requiring fewer buffer changes during long reads and writes.

- The new driver design should experience fewer data overruns and miss fewer X-ONs and X-OFFs.

- For terminal types that supply character (or TTY) delays, all carriage returns, line feeds, and form feeds will have the appropriate delay following the character. (The old driver, HIOTERM0, ignored these characters imbedded in user records.)

- An "ADCC Failure" routine has been incorporated into the new driver to handle software problems (this eliminates the possibility that the ADCC software will cause a system failure 208 or 209). The information about the failure or problem will now be dumped to a disc file, as done by the ATP.

- Speed sensing up to and including 9600 baud (with the exception of 150 baud) is now available on the ADCC.

- The "Idle Window" where characters may be lost will no longer exist. The ADCC hardware will continue reading data even while software is setting up the next operation. Control characters, however, will still cause the appropriate action; this is not a type-ahead operation, which implies that a person can type to the ATP/ADCC drivers, which will capture and process the information even though the buffers are full.

- And finally, in addition to the the above, a new product, the Workstation Configurator, which allows creation of unique terminal types by customers, will now work for ADCC ports running under the new driver (either HIOTERM2 or HIOASLP2).

Implementation Difficulties

Although the new driver is designed to be fully compatible with the old one, some differences are unavoidable. In general, where differences exist between the old ADCC and ATP drivers, the new ADCC driver will behave as the ATP driver does.

The following items are known differences that may cause customer problems:

- Subtypes 4 and 5 are not supported; they may exist in the I/O configuration (to allow regression to MPE IV) but are mapped into subtypes 0 and 1. (It was determined that subtypes 4 and 5 were being used simply so that speed sensing could be done at 9600 baud with the ADCC; because speed sensing at 9600 baud is now a standard capability, there is no longer a need for those subtypes.)

- The new driver does not speed sense at 150 bps; the :SPEED command must be used. It does, however, sense line speed and parity at line speeds of 110, 300, 600, 1200, 2400, 4800, and 9600 bps.

- Type 32 character printers must use the HIOASLP2 driver rather than the terminal driver (HIOTERM0 or HIOTERM2). (The name change is handled by the conversion program.)

- The new driver will not operate half duplex modems. (The old driver did not support half duplex modems either, but they did work.)

- The FOPEN IOQ on a modem port (subtype 1, 5, or 15) is processed in the manner of the ATP (HIOTERM1) driver; that is, the FOPEN IOQ is not returned until the modem handshake is completed.

- HIOTERM0 resets the ENQ counter only on a read or when the terminal type was set. The new driver will reset the counter on only set port protocol; that is, parity, speed, binary and set terminal type control operations (performed by FCONTROL).

- Parity is handled differently now. In HIOTERM0, parity checking always had to be enabled after a call to the driver to set parity. For example, when the following sequence

```
FCONTROL 36      ** Set Parity of A **
FCONTROL 24      ** Enable Parity Checking of A **
FCONTROL 36      ** Set Parity of B **
```

  was used with HIOTERM0, the parity of A would get checked, while the parity of B would not. In the new driver (HIOTERM2), the above sequence would result in the parity of B being checked also, because now, when parity checking is enabled, it will remain enabled until it is disabled (using FCONTROL 23).

IMF Performance Improvements

By Harold F. Seunarine

The Information Networks Division of Hewlett-Packard has made three enhancements to IMF/3000 Pass-Thru, all of which directly improve response time and increase throughput.

Improvement 1

Making use of the modified data-tags feature of the HP 2624A/B terminal has enabled us to reduce the time (up to several seconds per transaction) it takes to read a screen with a large number of unprotected fields (e.g., TSO/SPF editor).

Improvement 2

With the previous version of Pass-Thru, hitting a PF key involved up to four keystrokes. Now, seven of the HP terminal softkeys are user definable, and frequently used PF keys can be effected in a single keystroke.

The user may build a file called PTCONFIG (file equations allowed), which defines the PF/PA key to be associated with the HP terminal softkeys f2-f8. Softkey f1 is reserved by Pass-Thru to access all the other PF/PA keys not defined by the user.

The following is an example of a PTCONFIG file:

```
F2:PF2
F3:PF5
F4:PF6
F5:CLEAR
F6:PF8
F7:PF7
F8:PA1
KET:10
```

When a Pass-Thru session is started, a search for a file named PTCONFIG is performed in the user's group and account; a file equation may be used. If no such file is found, the default (current) definition of the function keys takes effect.

Improvement 3

With the previous version of IMF/3000 Pass-Thru, the IBM host would frequently send a screen to IMF in more than one transmission (sometimes three or four). There was no mechanism to alert Pass-Thru that the latest transmission was a part of a multiple-transmission screen, a problem that was disconcerting to our customers.

Now, by using the keyboard enable flag (which is sent by the host as an indicator of the final screen transmission), the Pass-Thru will not repaint the screen for intermediate transmissions; this improves the perceived response time. In the case where the host never sends a keyboard enable, a keyboard enable timer may be specified so that the screen will be painted after the timer period expires. (If the host enables the keyboard on the first transmission instead of the last, the screen repaintings will still occur; therefore, this feature is host-application dependent. For example, TSO normally does use keyboard enable in this way, while IMS frequently does not).

This feature is enabled by specifying a keyboard-enable timer value in the PTCONFIG file (e.g., KET:10 sets the keyboard-enable timer for ten seconds).

RJE Offers Auto-Dial, Programmatic Access, Other Features

By Steve Bitondo

RJE provides batch communication facilities for the HP 3000. RJE emulates the operation of the IBM 2780 or 3780 workstation, allowing a user to send card image files to a host and receive card and line printer image files from a host.

RJE software offers new features and ease-of-use:

- Auto-dial capability.
- Programmatic control.
- Self-measured performance statistics.
- Improved handling of EOT characters.

Auto-dial Capability

RJE can dial a telephone number and make a switched-line circuit connection without operator intervention. The applicable CCITT recommendation is V.25 and the applicable EIA Specification is RS-366.

Hardware Requirements for Auto-dial:

- INP Model HP 30020B.
- Automatic Dialing Unit - Bell System 801C (touch-tone).
- Special Auto-dial Cable HP 30221G.

Software Requirements for Auto-dial:

- INP download file CSDBSC2.*group*.*acct* .
- CS version A.04.03, or later.
- MPE IV (or later) operating system.
- RJE version E.02.00 or later.

Programmatic Control

Programmatic control in RJE allows concurrently running application programs to submit command files for RJE to process. Before attempting to use this feature, read the new Remote Job Entry Manual (30000-90047).

Self-Measured Performance Statistics

RJE can report its own performance, offering an objective measure of throughput. Performance statistics include, but are not limited to:

- Number of records sent to the host.
- Number of characters sent to the host.
- Number of records sent to RJE.
- Number of characters sent to RJE.
- Number of RJLINE commands executed.
- Number of RJIN commands executed.
- Number of RJOUT commands executed.

There are a total of 18 statistics kept by RJE and these statistics are reported relative to the beginning of RJE execution.

Improved Handling of EOT Characters

RJE no longer processes the *count* parameter of the RJOUT, RJLIST, and RJPUNCH commands improperly. RJE previously counted EOTs received from the host to determine how many datasets to process, assuming each EOT was the end of another dataset. If the host system sent multiple EOTs back-to-back without data, this would throw the count off. Now RJE only counts EOTs that are preceded by data in determining when the *count* parameter is satisfied.

NEW PERIPHERAL SUPPORT

SECTION VII

Introducing the HP 2687A and 2688A Printers

By Donna Sidman

The HP 2687A and 2688A are two new laser printers adding to the page printer product line. Both models use the same print engine, but each has different capabilities due to its unique controller.

The HP 2688A printer is plug-compatible with the HP 2680A. It interfaces with the HP 2680A Delta Control System (DCS) and HP-IB controller; it contains all text and graphics capabilities of the HP 2680A. The HP 2688A text and graphics printer supports over 100 fonts of differing type sizes and styles. The software applications that will be supported are HPWORD, TDP/3000, HPSLATE, HPDRAW, IFS/3000 and IDS/3000, DSG, and EZCHART. Two HP 2688As will be supported on a Series 39, 40/42, 44/48, and 64/68. The HP 2688A printer should be configured with the same type, subtype, and driver as the HP 2680A (type 32, subtype 8, and driver HIOPPRT0).

The HP 2687A will be configured with a serial interface (specifically, an RS-232-C for an ATP or an ADCC; or an RS-422 for an ATP only). It will serve as a high-quality text printing device and will have a maximum of four fonts addressable. The software applications supported are TDP/3000, HPWORD, and HPSLATE. The planned supported configuration for the HP 2687A is as follows:

On Series 39, 40, and 42:	One supported via ADCC.
On Series 44 and 48:	One supported via ADCC, or two supported via ATP.
On Series 64 and 68:	Four supported via ATP.

The HP 2687A printer should be configured with type 32, subtype 14, and terminal type 18. For MPE V/E, the driver should be HIOASLP2 with an ADCC and HIOASLP0 with an ATP.

The HP 2687A printer is not supported as a system printer, and there is no protection against errors occuring on this printer.

The HP 2563A Printer

By Donna Sidman and Adrienne Kelly

The HP 2563A printer is plug-compatible with the HP 2608S printer; plug-compatible is defined as a capability requiring minimal to zero software changes for support on MPE (minimal meaning a different subtype number, etc). The HP 2563A incorporates many of the HP 2608S features, and also provides a compressed printing ability (16.7 cpi) and a bar code printing ability. Improvements have been made in both density and character set flexibility as well. For example, while the HP 2608S does not allow more than two character sets on one line and would defer any other character sets to the next line, the HP 2563A is capable of printing up to 16 character sets on the same line. (For additional comparative information, ask your sales representative to check the HP 2563A/293X Sales Reference Manual.)

User convenience features include easy forms alignment, ribbon cartridge, paper jam detection, an optional sound abatement attachment, and an optional passive paper stacker. Dual densities are provided, as well as graphics capability, multiple character sets, and turnaround document capabilities.

The HP 2563A can be configured with a parallel interface (HP-IB); it must have the same driver as the HP 2608S, and the same type (32) and subtypes (9,13). The configuration supports four HP 2563As on Series 39/4X/6X.

The HP 2563A can also be configured with a serial interface (specifically, an RS-232-C for an ATP or an ADCC; or an RS-422 for an ATP only). It must be configured with type 32, subtype 14, and terminal type 19-22. The planned supported configuration for the HP 2563A is as follows:

On Series 39, 40, and 42:	One supported via ADCC.
On Series 44 and 48:	One supported via ADCC, or three supported via ATP.
On Series 64 and 68:	Six supported via ATP.

Two New Tape Drives: The HP 7974A and 7978A

By Larry Zeitman

Hewlett-Packard is announcing the support of two new tape drives on the HP 3000: the HP 7974A and the HP 7978A. These tape drives, supported on the Series 39/4X/6X, are low cost replacements for the HP 7970E and HP 7976A respectively. Because they use the same software I/O driver, HIOTAPE2, they share the same software characteristics and features. The driver is supported on MPE V/E (G.00.00).

This article describes the features, configuration, and data rates of the HP 7974A and 7978A tape drives. A section on the operating restrictions of both drives follows. A tutorial describing how the new features operate for both tape drives is also incorporated for any interested readers.

Features

The features of the two new tape drives are as follows:

- Multiple Density Recording. The HP 7974A records at either 1600 characters per inch (CPI) or 800 CPI (800 CPI, a purchased option, is not currently certified), with a maximum burst transfer rate of 160 Kb per second. The HP 7978A records at either 6250 CPI or 1600 CPI, with a maximum burst transfer rate of 468 Kb per second.

- The HP 7974A operates at a speed of 100 inches per second (ips) when streaming and 50 ips when nonstreaming. The HP 7978A operates only in the streaming mode, at a speed of 75 ips.

- Online Diagnostics. Both tape drives have small windows that display the status of their operations. Interpretations/instructions to obtain the displays are given in The HP 7974A Magnetic Tape Subsystem Operator's Manual (07974-90000) and The HP 7978A Magnetic Tape Subsystem Operator's Manual (07978-90000). The owners of these drives can diagnose many problems themselves, which effectively cuts maintenance costs.

- Streaming and Start/Stop Operation. Descriptions of the operating modes are offered on the HP 7974A and HP 7978A in the "TUTORIAL" section. The HP 7974A is a start/stop drive designed to operate in the streaming mode at higher tape speeds for faster data throughput. A major feature of this drive is automatic selection of Start/Stop or Streaming modes. The HP 7978A should be used only for streaming.

- Immediate Response protocol for high performance. The FSETMODE intrinsic enables you to turn on or off the immediate response protocol for increased writing throughput (refer to the "TUTORIAL" section).

- The HP 7974A offers a read-ahead capability for increased reading throughput.

- Automatic recovery from many errors.

Configuration

To configure the drive into the system, run the SYSDUMP dialog. In the dialog, answer "YES" to "I/O CONFIGURATION CHANGES?"; after the "SOFTWARE CHANNEL #?" prompt, enter the indicated responses to the prompts for the HP 7974A tape drive:

```
TYPE? 24
SUBTYPE? 3 or 11  ** "3" specifies manual reply; "11" specifies auto reply **
RECORD WIDTH? 128
OUTPUT DEVICE? 0
ACCEPT JOBS/SESSIONS? NO        ** "YES" if the drive is to accept jobs **
ACCEPT DATA? NO                 ** "YES" if the drive is to accept data **
INTERACTIVE? NO
DUPLICATIVE? NO
INITIALLY SPOOLED? NO
DRIVER NAME? HIOTAPE2
```

The response is identical for the HP 7978A, with the exception of the "SUBTYPE?" prompt. After this prompt the reply is either a "2" for manual or a "10" for automatic.

Data Rates

The following statistics for the HP 7974A are specified maximums and cannot be used to predict performance in a system environment:

Density	Speed	Maximum Burst Data Rate
(cpi)	(ips)	(Kb/sec)
1600	100 (streaming)	160
1600	50 (nonstreaming)	80
800	100 (streaming)	80
800	50 (nonstreaming)	40

The following statistics for the HP 7978A are specified maximums (not to be used to predict system performance). (This tape drive cannot perform effectively in the nonstreaming mode.)

Density	Speed	Maximum Burst Data Rate
(cpi)	(ips)	(Kb/sec)
1600	75 (streaming)	120
6250	75 (streaming)	468

Operating Restrictions

THE HP 7974A. If the door on the HP 7974A has been opened during operation, the tape access operation is aborted and the following message appears on the console:

```
8:20/#S3/117/ LDEV #ldev DOOR OPEN.  DOOR MUST BE CLOSED TO OPERATE
```

If the HP 7974A door has been opened during a :STORE/:RESTORE operation, the operation will additionally write to the console:

```
    8:20/#S3/117/ WILL RE-STORE ALL FILES THAT WERE ON BAD REEL
```

At this point in the :STORE/:RESTORE program, the drive rewinds and waits for the operator to mount a new tape. The messages displayed on the console are:

```
    8:22/#S3/117/ THE CURRENT STORE REEL ON LDEV #ldev IS BAD
    8:22/#S3/117/ PLEASE TAKE IT OFF THE TAPE DRIVE
    8:22/#S3/117/ IT IS NOT PART OF THE CURRENT STORE TAPE SET
    8:22/#S3/117/ MOUNT NEXT STORE REEL(volumenumber) ON LDEV #ldev
```

Both the :STORE and :RESTORE programs operate on only the reel that the abort occurred on. Other application or system programs must be restarted when the door is opened during processing.

THE HP 7978A. When cold loading from an HP 7978A, you will want to do a RELOAD (with the ACCOUNTS option rather than the default NULL) to take advantage of the tape drive's streaming ability. You will then want to restore the files with :RESTORE @.@.@;FILES=*maxfiles* (remember to use the FILES and CREATOR options of :RESTORE if necessary; refer to ":RESTORE COMMAND ENHANCED").

We do not recommend doing a RELOAD with the NULL option because the MPE Initiator driver has not been optimized for keeping the HP 7978A in the streaming mode. Furthermore, you can't take advantage of disc caching until MPE is running.

Tutorial

Commands and data are transmitted from the CPU to the tape drive, with each block of data following a command, as shown in the following figure. The tape drive responds to the CPU by acting on the data block in the manner specified by the corresponding command:

```
 _________                                        ___________
 :       :                                        :         :
 :       :          Immediate Response            :         :
 :       : <------------------------------------  :         :
 :  CPU  :                                        :  TAPE   :
 :       :                                        :  DRIVE  :
 :       : ------------------------------------>  :         :
 :       :    command, data, command, data        :         :
 :       :                                        :         :
 :_______:                                        :_________:
```

The data is then written, in sequential form, to the tape from the device's internal buffer. Unrecorded areas, called interblock gaps (or IBGs), are left between the data blocks:

```
 ________________________________________________________
 \                  |       |                          /
  \                 |       |                         /
   \   DATA BLOCK   |  IBG  |     DATA BLOCK         /
   /                |       |                        \
  /_________________|_______|_________________________\
```

START/STOP OPERATION. The primary purpose of a tape drive mechanism is to move tape from one reel (called the supply reel), across a read/write head, and back onto another reel (called the take-up reel), at relatively fast speeds and in an extremely accurate path. To perform these functions, every tape drive has motors to move the reels, a read head, and some guidance hardware to keep the tape on track. In addition, there are control boards and interface cards, which convert the stream of data bits from the host into a pattern that characterizes the recording format. As the CPU and the tape drive communicate, the tape drive has to periodically stop the tape to allow for the verification of the past block and the initialization of the next block. There is a length of unrecorded tape, called the interblock gap, that is used when the drive decelerates and reaccelerates to write or read. The faster the drive accomplishes this starting and stopping routine, the less blank tape would be wasted between blocks. However, because it is difficult to accelerate and decelerate the reels quickly, tape drives have been built with capstan motors to control the speed of the tape over the read/write head and tape buffering mechanisms to provide reservoirs of tape that are easier to move quickly. Two common buffering mechanisms are "tension arms", which operate on extremely sensitive springs, and "vacuum columns", where tape is floated in long, narrow chambers. Where the maximum forward tape speed is around 50 inches per second (ips), tension arms are sufficient. Faster speeds, however, require vacuum columns. Sensors in the buffering mechanism tell the supply and take-up reel motors when they need to refill the tension arm or vacuum column reservoir. Tape drives that have these kinds of mechanisms are called "Start/Stop" drives. Start/Stop drives require anywhere from 5 to 20 milliseconds to stop and reaccelerate within the IBG.

STREAMING. If the tape drive controller is provided with enough room to hold more than one command (including the associated data blocks), then the drive could be acting on one command (putting the data on tape) while the host could be sending the next command with its data block. When the drive finishes with the first command it could then continue directly on to the next command without having to stop. This kind of operation, where the drive does not have to stop at the IBG, is called the "Streaming Mode". (This term has no relationship to the physical tape mechanism; instead it refers to the way in which the host talks to the tape controller, and the consequent effect on the movement of the tape.) If the host can keep supplying commands and data blocks to the tape drive quick enough, the drive stays in the streaming mode (it does not need to stop or start frequently), and performance is significantly improved.

The tape buffering mechanism is an expensive part of the tape drive. The HP 7978A has been designed without any tension arms or vacuum columns to reduce its cost significantly. It is called a "pure streaming device". A pure streaming device must ramp down slowly to avoid breaking or stretching the tape. When new data is transmitted,the drive must rewind to a point before the last written block, then ramp back up to speed gradually. This procedure is called the reposition cycle. It is quite time consuming, requiring hundreds of milliseconds. To maintain satisfactory performance, the HP 7978A was designed to allow creation of variable-length interblock gaps to give the HP 3000 as much time as possible to supply data to the drive. And MPE utilities were redesigned to keep the HP 7978A in the streaming mode.

AUTOMATIC SPEED ALGORITHM (FOR THE HP 7974A ONLY). The HP 7974A is both a streaming and start/stop device. The HP 7974 streams data at 100 ips. It begins in streaming mode by default and continues streaming as long as commands and/or data remain in the device's buffer. If the host does not supply data in time, the HP 7974A repositions itself and waits for data. If too many repositions occur, the device automatically shifts to stop/start mode (50 inches per second). When the HP 7974A's buffer accumulates data again, it will automatically switch to streaming mode. MPE utilities, `:STORE/:RESTORE`, and SPOOK5 have been enhanced to take advantage of the HP 7974A's streaming characteristic. You should observe the improved performance of utilities and applications programs that utilize the stream mode (depending, of course, on system load and system characteristics).

IMMEDIATE RESPONSE. Immediate response is another method of increasing performance for write operations. Immediate response interchanges the execution phase of a command sequence with its report phase. This prevents the HP 3000 from waiting on the drive to complete a request, thereby allowing the system to prepare the next request.

With immediate response enabled, the device responds with a status (CCE, CCL) immediately after the DMA transfer from a file's buffer (b, in Figure 1, below), in memory, to the tape drive's internal buffer. This allows another transfer to be sent to the drive while it is transfering data from the internal buffer to tape.

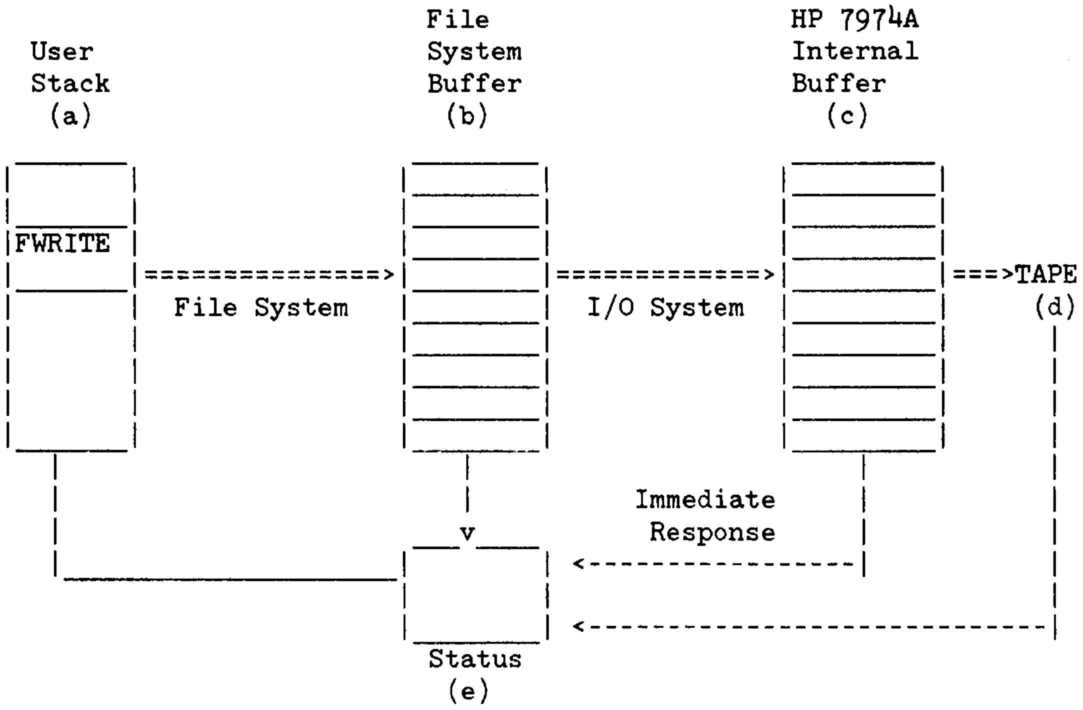


Figure 1

With immediate response disabled , the I/O system waits for the tape drive to post a full block of output to the tape before control is given back to the FWRITE intrinsic and a CCL or CCE is returned. When immediate response is enabled, a write to tape proceeds in this manner (refer to Figure 1 above):

1. When an FWRITE is issued, the File System moves a logical record from the users stack (a) to a file system buffer (b), and returns a status (i.e. a CCL or CCE) to the user program so that the program continues execution.

2. When a file system buffer becomes full, it has a buffer size that is equal to one physical block; the physical block is transferred to the HP 7974A's internal buffer. (The maximum record size is 16,384 bytes.) The status of this transfer, again, is put into the status area (e). This is called immediate response.

3. The device transfers a physical record from the internal buffer to the tape, and the result of that transfer is also placed in the status area (e).

It is possible that a user program might receive an immediate response from a write to tape and attempt to close the tape file before the status from the physical write has been reported to the user program in the status area. To ensure that the program receives the status of the last physical write to tape, the application program should issue two "end of files" (EOFs) to the file. The program will wait at the second EOF until all the records have been physically written on the tape. If it is critical to know the status of each write to a tape before the next one is issued, the application program should disable the immediate response.

Immediate response, which is enabled by default, can be turned on or off with the FSETMODE intrinsic:

FSETMODE (***filenumber*** , ***modeflags***);

modeflags Bit (14:1) - Critical Output Verification.

0 - Immediate response is enabled (default).
1 - Immediate response is disabled.

READ-AHEAD CAPABILITY (FOR THE HP 7974A ONLY). The read-ahead capability increases the reading throughput of the HP 7974A. This allows continuous transfer of data after the end of the current record into the device's internal buffer. This transfer continues until either another command is received, the internal buffer is filled, a read error occurs, or an "End or Interrupt" (EOI) condition occurs.

The read-ahead operation does not change the logical position of the tape for the current request. Operating system requests are processed against the internal buffer until the operation exceeds the limits of the internal buffer. When this occurs, the device will again fill the buffer with data (repositioning itself as required, depending upon the request that caused the buffer to be filled). This feature is always enabled; the application programs will see no difference in operation except that the tape drive will respond faster to read requests.

AUTO RECOVERY. All read, write, and space-type (forward-space record, backspace record, etc) commands use auto error recovery. For read or space-type operations, the drive will attempt seven retries before classifying the error as unrecoverable. For write commands, the drive will attempt three retries at the same place on the tape. If these are unsuccessful, the drive will attempt fifteen more retries. With each of these fifteen attempts, the drive will first execute a "gap tape" command (which instructs the tape drive to erase approximately 3.5 inches of tape), and then it will retry the record. The position of the tape will be at the end of the last attempted retry of the record and status will be returned to the application.

1/4 Inch Cartridge Tape Driver Change

By Satish Janardan

On MPE V/E, the driver for the HP 7911P, 7912P, and 7914P Integrated Cartridge Tape Units has changed, allowing the user to abort all pending I/O requests (via `:ABORTIO`). This new feature enables the user to abort the VINIT subsystem (or SLEUTHSM) formatting process for tapes.

KSAM Forms Files Not a Thing of the Future

By Ron Harnar

Several software releases ago, VPLUS began transition to a non-KSAM (Keyed Sequential Access Method) file format:

- Currently, you cannot create new KSAM forms files with FORMSPEC or REFSPEC.
- Soon, you will not be able to modify KSAM forms files with FORMSPEC or REFSPEC.
- You will always be able to open and access KSAM forms files with VPLUS intrinsics.

Users Affected By KSAM Changes

This change will only affect users who currently have KSAM forms files or reformat files. Currently, files created with FORMSPEC or REFSPEC do not use KSAM (Keyed Sequential Access Method). Because of the disadvantages of KSAM, most VPLUS users have already converted to non-KSAM format.

The new non-KSAM format provides faster access and more efficient use of system resources, and can be maintained without KSAMUTIL.

How to Convert from KSAM to non-KSAM files

The conversion utility `CONVERT.PUB.SYS` is part of every new software release. When run, CONVERT asks for the name of the KSAM file to convert, and the name to be given the new file. CONVERT can be run in batch mode.

For more details, refer to the HP Data Entry and Forms Management System (VPLUS/3000) Reference Manual (32209-90001).

Enhancements to SPL/3000

By Jon W. Henderson and Dennis Handly

The purpose of this article is to acquaint the reader with enhancements which have been made to SPL/3000 in version A.08.04. These enhancements are:

- Methods to improve, at compile time, the reliability of code executing in split-stack mode.
- The ability to use the MOVE and SCAN statements as integer functions.
- Changes to listing format which improve the readability of programs.
- Miscellaneous other enhancements.

Q-Relative and S-Relative LRAs in Split-Stack Mode

Aside from explicit ASSEMBLEs , Q- and S-relative LRAs are generated primarily in four cases:

- Allocation of storage for indirect arrays declared in procedures.
- When the address of an indexed element of a local array or pointer (i.e., @<*variable*>[<*index*>]) is used on the right-hand side of an assignment statement.
- OWN variable declaration.
- Multiple assignments of LONG variables (i.e. L:=L1:=L2).

NOTE

The following enhancements have caused these words to be added to SPL's list of reserved words: DATASEG, MOVEX , SPLIT , and WITH.

Programs containing these identifiers will have to be modified and recompiled.

Since the "distance" between DB (LRAs are always calculated in terms of DB) and Q or S may have changed as a result of the call to EXCHANGEDB , the offset loaded by the LRA may be incorrect, causing the wrong memory location to be modified. (Indirection and indexing compounds the problem.) A new $ option and a new procedure option have been added, which check for and flag as errors any Q- or S-relative LRAs that are generated while the options are in force. When $SPLIT is encountered, the compiler begins the checking described above. The "$" options are of the form:

```
$SPLIT
$NOSPLIT
```

The $NOSPLIT option disables the checking. Thus, an area of code (usually between calls to EXCHANGEDB) can be checked for "dangerous" code.

In some instances, such as local variable allocation within a procedure, the code can be generated before any $ options are processed. Therefore a new procedure option, SPLIT , is available. As with other procedure options, SPLIT is recognized before any code is generated. Also, it remains in force for the entire body of the procedure and cannot be reset by $NOSPLIT .

The form of the declaration is:

```
PROCEDURE P;
OPTION SPLIT;
```

Uninitialized Indirect Cells in Split-Stack Mode

Another common mistake is the use of uninitialized indirect cells (pointers). For example:

```
BEGIN
INTEGER I;       <<DB+0>>
DOUBLE D;        <<DB+1>>
ARRAY A(0:99);  <<DB+3>>

A(99):=99;

END.
```

At program load time, DB+3 will be initialized with the offset in secondary DB of the beginning of A . However, if a call to EXCHANGEDB is done, DB+3 in the new data segment will most likely contain a spurious value. If this happens dereferencing DB+3 may result in a bounds violation; or some completely random memory location in the machine may be modified by the assignment. To help this situation, two new structures have been added to the SPL syntax: the DATASEG declaration, and the WITH statement.

DATASEG DECLARATION. The DATASEG declaration limits the variables declared to certain explicit DB-relative offsets. The BNF form of the declaration is:

```
<dataseg declaration> ::= DATASEG <dataseg declarator>
                          BEGIN
                          <dataseg declarations>
                          END;

<dataseg declarator> ::= <dataseg name> = <dataseg #>

<dataseg #> ::= <integer constant> !
                <integer constant expression> !
                <equate>

<dataseg name> ::= <SPL identifier>

<dataseg declaration> ::= <SPL variable declaration> [= <dataseg offset>]
```

```
<dataseg offset> ::= <dataseg name> [+ ! - <offset>] !
                     <dataseg variable> [+ ! - <offset>]

<offset> ::= <integer constant> !
             <equate>
```

For example:

```
DATASEG D = 77
BEGIN
INTEGER I;     <<OFFSET = 0>>
REAL R = D+5, <<OFFSET = 5>>
LONG L = R+2; <<OFFSET = 7>>
ARRAY A(0:5); <<OFFSET = 1>>
END;
```

Note that the BEGIN and END are not required if there is only one element in the DATASEG. Only simple variables, arrays, and pointers are permitted as declarations in DATASEG. No GLOBAL, EXTERNAL, or OWN declarations are allowed. The declarations <var> = <dataseg name> and <var>=<dataseg name> +0 are equivalent. A variable declaration without an offset will be assigned the next available offset.

WITH STATEMENT. This statement is used in conjunction w.ːh the DATASEG declaration. The WITH statement is a syntactic device which ensures that only split-stack mode compatible code is generated, and that only those variables which have been declared inside the DATASEG are used. All of the checking done by $SPLIT is performed, as well as the above-mentioned checking of the variables used.

The variables used inside a WITH block must have been declared inside a corresponding DATASEG declaration, or be Q- or S-relative. An exception is a MOVEX between data segments, which is the only form of MOVEX allowed inside a WITH statement. In this case, the variable(s) used may have been declared inside any DATASEG declaration.

NOTE

Q- and S-relative variables, particularly pointers, may have erroneous values upon entry to a WITH block. No checking can be done for this.

The form of the WITH statement is:

```
WITH <dataseg name> DO
BEGIN
.
.
.
<dataseg-relative variables>
.
.
.
END;
```

The BEGIN and END are optional if there is only one statement in the WITH block. The call to EXCHANGEDB must be explicitly issued preceding and following the WITH block.

NEW INSTRUCTIONS. A new statement, MOVEX , has been added. This statement causes, when necessary, the MFDS , MTDS , and MDS operation codes to be generated. This allows high-level split-stack code to be written, thereby increasing its reliability. The format of a MOVEX statement looks as follows:

```
MOVEX (<D> [,<OFF>]) := (<S> [,<OFF>]) ,(<L>) [,<SDEC>] ;
```

D,S Either DB-relative pointers (for MFDS and MTDS), DATASEG or DATASEG-relative identifiers (for static XDS moves), or integer expressions (for dynamically calculated XDS numbers). In the latter case, DATASEG-relative identifiers are not permitted in the expression.

OFF Optional. The beginning offset into the XDS . It can be either a constant or an integer expression composed solely of identifiers that are valid within any containing WITH or $SPLIT statement. An offset is not permitted when the pointer is DB-relative (as opposed to DATASEG-relative).

L The number of words to be moved (not bytes).

SDEC (optional) The same as in standard MOVEs , except that its maximum value is seven.

For example:

```
MOVEX (D, 9) := (D1, I + J), (K), 6;<<Static data segment #s>>

MOVEX (99, I+J/2) := (K*M, L), (99);<<Dynamically calculated>>
```

Note that in the case of a MOVEX between variables declared in the same DATASEG , ordinary DB-DB MOVE code is generated.

The basic philosophy behind MOVEX is to allow programmers a high-level manner in which to access the MTDS , MFDS , and MDS operation codes.

CAUTION

The normal checks and limitations that apply to standard users in MPE are bypassed in privileged mode. It is possible for a privileged mode program to destroy file integrity, including the MPE operating system software itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of privileged mode code. This service, which is not provided under the standard Service Contract, is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modification of the MPE operating system software.

MOVE/SCAN Enhancements

The MOVE and SCAN statements have been enhanced to behave, when desired, as arithmetic functions returning the number of bytes/words moved. This enhancement is discrete from the MOVEX enhancement; MOVEX cannot be used in this context. The existing syntax of MOVE and SCAN remains intact; existing programs will recompile without incident. Both arithmetic statements can be used anywhere an integer expression would be appropriate.

For example:

```
    I := MOVE P:=P1, (<<LENGTH>>);    <<Trivial, since fixed length >>

    IF P(MOVE P:=P1 WHILE ANS) = "xyz" THEN . . .;

    MOVE P:=P1, (SCAN P1(SCAN P1 UNTIL " ") UNTIL " ");
```

Since the arithmetic MOVE and SCAN may be used in expressions, no SDECs are allowed. This is due to both the code generated to calculate the return value, and the desire to minimize the possibility of corrupting the stack. Also, carry/condition code is not saved for the SCAN and MVBW (MOVE .. WHILE) instructions.

Listing Enhancements

The listing enhancements include: $CONTROL AUTOPAGE , decimal-format line numbers, and enhancing the $CONTROL MAP option.

$CONTROL AUTOPAGE. This enhancement causes a page eject whenever a procedure declaration is the first token found on a line. If the declaration is preceded by COMMENT or << , no page eject will be issued; however, if the embedded "declaration" occurs after the first line of the comment, one will be issued. Similarly, any documentation (procedure headers, etc.) placed before the procedure declaration will appear on the preceding page.

DECIMAL-FORMAT LINE NUMBERS. Line numbers now appear in the format XXXXX.XXX , with leading and trailing zeroes suppressed. Only the listing file is affected by this enhancement. Any programs which operate on the line numbers in SPLLIST, and which depend on the old format, will have to be modified to accept the new format.

$CONTROL MAP. In the Identifier Map, reference parameters are flagged with an " (R) ". If the warning "SORT TABLE OVERFLOW " occurs the identifier map will be printed in sections, each section alphabetized. All the identifiers will appear.

Miscellaneous Enhancements

The following additional enhancements have been made:

- The number of local arrays declarable in a procedure has been increased from 64 to 127.

- If the special character "!" is encountered outside of a comment, definition, or string, the rest of the source line following the exclamation point will be regarded as a comment, i.e.:

  ```
  I := 0; ! This is a comment
  ```

- Hitting Y^C during a compilation will cause the number of errors and warnings generated so far, as well as the current line number, to be displayed.

- The INFO parameter can now be used to pass $CONTROL commands to SPL. The format is:

  ```
  :SPL <files >;INFO="$CONTROL MAP $CONTROL ADR,NOLIST "
  ```

  Any number of commands may be concatenated. The commands passed will be identified in the listing by a sequence number of "INFO ". Any $CONTROL commands already in the source will override the passed commands.

- Assemble statements can now be typed. They can be used where an expression of that type is allowed. For example:

  ```
  I:=Integer Assemble (PCAL %0) + 2;
  ```

Changes to Pascal/3000

By Sue Kimura

The Pascal/3000 released on the G.00.00 version of MPE V/E represents the most significant development since the release of the original version (A.00.00). A major effort has been made to reduce the number of string-related problems.

A major enhancement, adding the capability of symbolically debugging a program under the HPToolset environment, has been completed. A new compiler directive $SYMDEBUG$ directs the compiler to put additional information into the User Subprogram Library (USL) file which in turn is added to the program file, by the segmenter. The USL file may also be prepared with the NOSYM option if the user does not want the symbolic information in the program file.

When a program compiled with $SYMDEBUG$ is run under HPToolset, the contents of variables, both simple (integer, real, or char) and structured (array or records), may be displayed by referencing the variable name, or in the case of a pointer variable by dereferencing the pointer variable. Simple variables may be modified by referencing the variable name; structured variables may be modified by referencing another variable of the same type. Breakpoints may be set using statement numbers, and the dynamic flow of a program may be symbolically traced. The use of symbolic debug should eliminate the use of MPE DEBUG or the use of numerous write statements used just for debugging.

The user should note that in order to implement this capability two more extra data segments are used by the Pascal/3000 compiler in addition to the four already being used. The user's computer configuration may be affected by this change.

To facilitate HPToolset's symbolic debug and program key functions, changes have been made to the source listing format. Statement numbers now appear in the first column and editor line numbers in the second column. Statement numbers are reset to zero at the start of each level one procedure. The statement number of the first executable statement of a procedure or function or the outer block no longer necessarily starts at zero, since labels and identifiers (procedure names, constant names, type names, and variable names) are now also assigned statement numbers.

A $IF compiler directive, which allows conditional compilation of source text, has been added to the Pascal/3000 compiler. The format of the $IF directive is:

```
$IF 'xxxx'$
   ** If-statements **
$ELSE$
   ** Else-statements **
$ENDIF$
```

The 'xxxx' is a literal string which is set to TRUE or FALSE by a $SET 'xxxx = TRUE' or a $SET 'xxxx = FALSE'$ directive before the program statement. If 'xxxx' is TRUE , the if statements are compiled. If 'xxxx' is FALSE , the else statements are compiled. The string in the $IF directive may contain a complex expression using OR , AND ,or NOT (logical operators), and parentheses, as in the following example:

```
$IF 'cond_1 AND (NOT cond_2)'$
```

Another new compiler directive is the $FONT directive which allows the user to set the primary and secondary character sets for a source listing on an HP 2680A page printer. The format of the $FONT directive is:

```
    $FONT '<unsigned_integer>, <unsigned_integer>'$
```

The first integer represents the set number of the primary font and the second integer the set number of the secondary font. The user may use these fonts in the $TITLE '<string>'$ directive or in a comment. To switch to the secondary character font, a N^C character is inserted in the '<string>' or comment. To change back to the primary character font, a O^C is inserted.

In addition to these new compiler directives, changes have been made in file I/O. A new HP standard function, LASTPOS(*f*), has been added to Pascal/3000. The function LASTPOS(*f*) returns the integer index of the last component which has been written on *f*. This differs from MAXPOS(*f*) which returns the integer index of the last component of *f* which may ever be accessed. The file *f* must be opened for direct access. With the addition of this new function, the HP standard definition of EOF(*f*) , when *f* is in the direct access state, has been changed so that EOF(*f*) returns true when the current position of *f* is beyond LASTPOS(*f*) . The Pascal/3000 implementation of EOF(*f*) has always conformed with this definition.

Changes have been made in the implementation of the READ and WRITE statements. A modification to the ANSI standard affects READ and WRITE statements using multiple *v* parameters. According to the new standard the actual file referenced remains fixed through the entire statement. For example, in the statement READ (*f*, *v1*,...*vn*) the actual file referenced is now determined once and a read is done for (*v1*,...,*vn*) from that file reference.

A change to the HP Standard has been made with respect to reading PACs (Packed Array of Char), and strings. If a variable of type PAC is being read and EOLN is true, a blank-filled PAC is returned. In the case of a string, a zero length string is returned. The user must use READLN to advance beyond the current line.

A performance improvement has been made in reading PACs and strings from text files. An increase of up to 40 percent in the speed of reading PACs and strings from text files may be noticed.

Finally, two notable changes have been made to the error and warning messages issued. First, the message " VARIABLE ACCESSED BUT NOT INITIALIZED " is now a warning and not an error. An effort has been made to detect more occurrences of uninitialized variables and this effort will continue in future versions of the compiler. Secondly, the occurrences of the warning " BYTE TO WORD CONVERSION " have been greatly reduced.

Customizable HPMenu Program Gives Access to HP 3000 Applications and Commands

By Mona Matsumoto

HPMenu is the easy-to-learn menu building facility that makes access to commands and programs easier for office personnel. HPMenu protects the user from complex command syntax by presenting choices on the screen. Each user can have his own menu(s).

HPMenu can access up to 100 selections, which can include:

- Other Menus.
- Interactive Office software.
- MPE commands.
- Most other HP application software (except MM/3000 and PM/3000).
- Customer-written software including :STREAM files, or Third Party software (except those which run with the master/slave terminal setup like MPD applications).

Users can customize their own menus. Tailoring the menus is simple for non-programming office workers.

Users can also be restricted to use of a particular menu, shielding them from functions not in the menu. Access to the MPE operating system or the customization of menus can be restricted.

Feature Summary

The features of HPMenu include:

- HPMenu protects the user from having to interact with the MPE operating system, or having to use complex command syntax.
- Any menu choice can be selected in three keystrokes or less.
- Programmers can utilize a parameter table to pass information from the user to the system. Data processing stream files and applications requiring parameters to be passed to them can be included in menus.
- A utility allows copying User Defined Commands (UDCs) into menus to make installation easier.
- Menus can be changed easily, or alternately, customization ability can be "turned off" for added security.
- On-line help for using and customizing HPMenu is available.

Hardware and Software Requirements

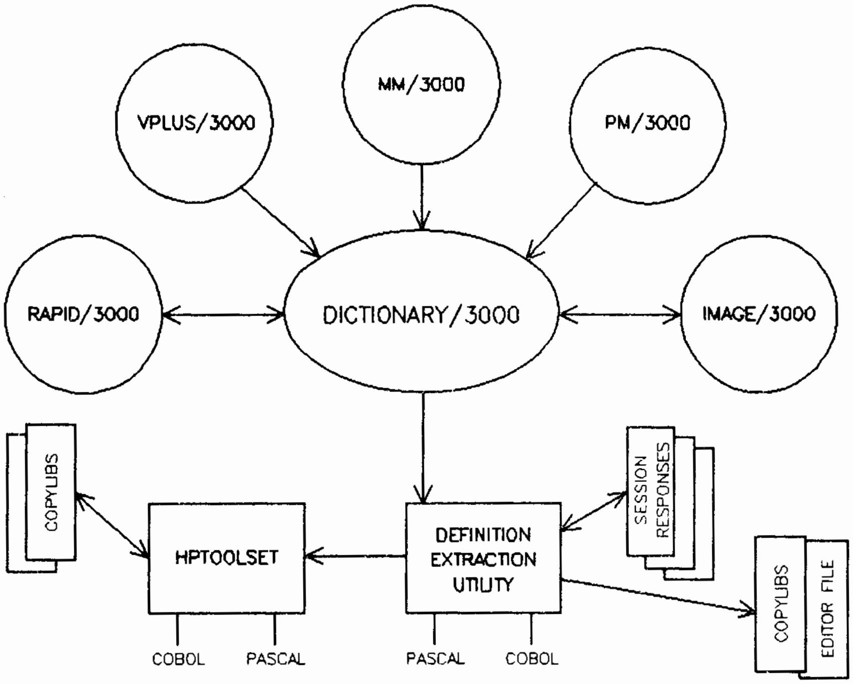
HPMenu must operate with VPLUS/3000 version B.03.03 or a subsequent release (part of the Fundamental Operating Software), and any currently supported block-mode terminal. Point-to-point asynchronous connections, DSN/DS, and multipoint terminal software connections are supported by Hewlett-Packard. For details, refer to the new HPMenu Reference Manual (32112-90000).

DICTIONARY/3000 Enhancements

By Susan Gennrich and Marty Miller

The A.02.00 release of Dictionary/3000 includes significant enhancements that integrate this product with other HP products. Dictionary/3000 is currently used for documentation, control of a data base environment, and as the the source of definition and location information for the RAPID/3000 products. New features and utilities allow you to:

- Extract Dictionary definitions as COBOL ENVIRONMENT and DATA division declarations (DICTCDE utility).
- Extract Dictionary definitions as Pascal TYPE and VAR declarations (DICTPDE utility).
- Load VPLUS form definitions into the dictionary automatically (DICTVPD utility).
- Link Dictionary to MM/3000 and PM/3000.



Dictionary/3000 Integration

The Enhancements

The COBOL and Pascal extract utilities, DICTCDE and DICTPDE, will generate source code that corresponds to files and data elements defined in the dictionary. The COBOL utility generates ENVIRONMENT and DATA division declarations. The Pascal utility generates TYPE and VAR declarations. Both utilities will generate declarations corresponding to IMAGE, KSAM, MPE and VPLUS files. Declarations for standard parameters used in accessing IMAGE and VPLUS files can also be generated. Both utilities will handle arrays, tables, and alternate record formats.

The COBOL and Pascal utilities can be used in two ways. First, they can be run stand alone in both session and job modes. The definitions generated with this approach will go into a copylib for COBOL and an editor file for Pascal. When running in session mode, responses can be saved in a file and used again as responses in a later job or session. The second method of running the extraction utilities is through HPToolset, which has been enhanced to call the extraction utilities from within its own environment. The definitions generated in this instance are written directly into the file or copylib being edited.

Included on this release is a utility, DICTVPD, to automatically load form definitions into the dictionary from existing forms files. This utility has a user interface similar to the DICTDBD utility that loads existing data base definitions into the dictionary.

A link between Dictionary/3000, MM/3000, and PM/3000 has also been added. Now the customizer will automatically update Dictionary to reflect changes made to the MM/3000 and PM/3000 systems. This allows MM/3000 and PM/3000 customers to take advantage of the reporting capabilities of Dictionary/3000, Report/3000, and Inform/3000.

Special Notes

The structure of the Dictionary has changed to allow for the enhancements. This means that existing users of Dictionary/3000 will need to run the DICTINIT program to reinitialize existing dictionaries to the new dictionary structure.

The DICTDBM program has been altered to reflect the new enhancements. Existing Dictionary/3000 users will notice a slight difference in the user interface, as additional commands have been added to allow for integration with COBOL and Pascal. However, the general use of the product is essentially unchanged.

A new edition of the Dictionary/3000 Reference Manual (third edition, 32244-90001), which reflects the new enhancements, is available. The current user base is encouraged to review this manual upon receipt to note the changes made in the user interface. An updated classroom course and an updated self-paced training course are also available.

A new message catalog now contains all of the error messages and user information strings. This catalog file (DICTCAT.PUB.SYS) is required to run any of the utilities, including DICTDBM.

Creating Family Relationships in Formspec And Other VPLUS Enhancements

By Nancy Ofslager

In response to customer requests for a method to re-establish family relationships after a copy operation, a new command has been added to Formspec. The new command, entitled RELATE, can be used to create family relationships between forms that have certain properties.

When designing the family-form capability, the engineers foresaw the need for users to dissolve family relationships. They felt that the copy command would be the best way to implement this idea. Since that time, however, it has become apparent that users also have a need to copy family forms and still maintain the original relationships. The RELATE command meets this need. This command brings the ability to link forms that may or may not have been previously related.

From Formspec's Main Menu, the RELATE command can be initiated by entering an "R" into the selection field, along with the names of the proposed son and father into the last two fields. The RELATE command can also be enacted from a batch mode command file using the following syntax:

```
RELATE son-form TO father-form
```

Both the proposed son and the proposed father must meet certain criteria for the RELATE command to execute. First of all, the forms must exist and their screen designs must be identical. The forms must also be distinct forms; that is, one form cannot be both the parent form and the child form. Furthermore, the son form cannot already be a son or father form. If these conditions are met, the forms can be related. Thus, it is possible to establish a relationship between forms that were never previously related.

Users should note that if the father-to-be is currently the son of another form, then its father will be designated as the actual father-to-be. For example, suppose an existing forms file contains three forms named FATHER, SON, and ORPHAN, where FATHER and SON already maintain a relationship. From the Main Menu, it is indicated that ORPHAN should become a son form of SON. Assuming that the forms are able to be related, Formspec will actually create a relationship with FATHER as the father form and ORPHAN as the son form.

RELATE Command Error Messages

The following new error messages have been added to Formspec for the RELATE command. Please add these messages and actions to the error section of the Data Entry and Forms Management System (VPLUS/3000) Manual (32209-90001). It will be updated at a later date.

Error	Message	Cause/Action
791	SON CANNOT HAVE THE SAME NAME AS THE FATHER	Same form name was specified as the father and the son.
792	FORM IS ALREADY SON OF THAT FATHER	The forms are already related.

793	SON FORM IS ALREADY RELATED TO ANOTHER FATHER	A son cannot have more than one father.
794	FORM SPECIFIED AS SON CAN NOT HAVE ANY SONS	Cannot relate a father form to another form.
795	FORM LAYOUT OF SON MUST BE THE SAME AS THE FATHER	The screen designs must be identical, including field tags.
796	Reserved for future use.	
797	CORRECT SYNTAX IS: RELATE *son-form-name* TO *father-form-name*	Arguments to the batch mode command were mis-typed.

Support of New Terminals

Support of the recently announced HP 2625A and HP 2628A terminals can be found in the new B.03.20 version of VPLUS/3000. Support of the HP 150 is also incorporated in this release. The new data capture terminal, the HP 3081, has been supported since VPLUS version B.03.15.

Other Enhancements

The following enhancements were incorporated in VPLUS version B.03.15. Some of this information is documented in the May 1983 manual update and is offered here to highlight the new features.

REMOVING FUNCTION KEY LABELS. Global or form function-key labels defined in Formspec can now be removed. This is accomplished by entering an "N" in the function key label field the Form or Global Menu from which the labels were defined. Formspec will now place either an "X" in the function-key label field to indicate that labels have been defined, or a blank to indicate that labels have not been defined. Users will continue to enter a "Y" in the function-key label field to modify or add function-key labels.

ENABLING THE AIDS, MODES, AND USER KEYS IN APPLICATIONS. Many applications have the need to access the AIDS, MODES and USER keys.

In order to allow form-by-form enabling and disabling of these keys, a new bit has been defined in the SHOW'CONTROL word of the communications area. The eighth bit (relative to 0) of the SHOW'CONTROL word now has the following meaning:

0 = Do not enable the AIDS/MODES/USER keys.

1 = Enable the AIDS/MODES/USER keys. These keys are enabled after a form is displayed on the screen and remain enabled until bit 8 is reset to 0.

RUNNING FORMSPEC ON PORT 2 OF THE HP 2626A TERMINAL. VPLUS has been enhanced to allow Formspec to be run on Port 2 of the HP 2626A terminal. As the HP 2626A terminal does not indicate to VPLUS which port is being used, local form storage and execution from Port 2 are mutually exclusive. This is because either the window and workspaces must be assigned to a single port for local form storage, or the configuration should be left unchanged for execution from Port 2. Because of the number of enhancement requests received, the decision was made to support Formspec on Port 2 and discontinue Formspec's use of local form storage on the HP 2626A terminal. User applications have always been able to run on Port 2 of the HP 2626A terminal by initializing the FORM'STORE'SIZE word in the communications area to "-1".

VPLUS VERSION COMPATIBILITY. The following is a VPLUS/3000 version compatability chart. It indicates the possible forms files and intrinsic version levels, along with the compatible and incompatible combinations.

		INTRINSIC VERSION LEVEL	
		PRE- B.03.03	B.03.03 & BEYOND
FORMS FILE VERSION LEVEL	PRE- B.03.03	COMPATIBLE	COMPATIBLE
	B.03.03 & BEYOND	VOPENFORMF WILL RESULT IN INTEGER OVERFLOW	COMPATIBLE

As the chart indicates, once VPLUS forms files have been converted to version B.03.03 or later, they are not compatible with VPLUS intrinsics of the pre-B.03.03 level. Please note: simply accessing the forms file with ENTRY or an application does not convert the forms file. In order for the forms file to be converted, FORMSPEC must be run against the forms file, with "CONTINUE" entered in the Adjust Menu. It is not necessary to convert the forms file to run on B.03.03 or later versions. The conversion process only is required if you wish to modify the forms file via FORMSPEC of the B.03.03 or later level.

VPLUS FILE EQUATIONS. VPLUS checks for three file equations during execution. The use of these file equations provides the user with a way to override the defaults used by VPLUS. The VPLUS intrinsic VOPENTERM (used by FORMSPEC, ENTRY, and REFSPEC) searches for a file equation named "A264X". VOPENTERM will attempt to open the device referenced by "A264X" to display VPLUS forms and accept their associated input. For example:

```
:FILE A264X; DEV=28
:RUN ENTRY.PUB.SYS
```

will cause logical device 28 to be opened by VOPENTERM, and subsequent screen displays by VSHOWFORM will appear on LDEV 28. In this example, LDEV 28 must be a logged off terminal in an available state.

User applications can make use of this feature by using the *termfile* parameter of VOPENTERM and a file equation. If the second parameter to VOPENTERM contained the value "TERMINAL", the following file equation would cause logical device 28 to be opened for forms display:

```
:FILE TERMINAL; DEV=28
```

Display statements and abort messages will be displayed on the command terminal (i.e. the one that initiated the application). This would be the main reason to use this feature, as abort messages are frequently impossible to read because they are displayed in the unprotected fields of a VPLUS application.

The second file equation that VPLUS searches for is FORMLIST. VPRINTFORM uses FORMLIST to list forms (as in ENTRY). FORMLIST defaults to the system printer. If, however, the user wishes to save the list file on disc, the following file equation will do just that, creating a file named FORMLIST that contains the form listing.

```
:FILE FORMLIST; DEV=DISC; SAVE
```

If your system has both a laser printer and a line printer (default), the following file equation will send the form listing to the laser printer:

```
:FILE FORMLIST; DEV=PP
```

The last file equation used by VPLUS deals with a summary that lists the forms file from batch mode Formspec. The file name for this file equation is FORMOUT, which can be used as FORMLIST in the previous example.

Summary

The B.03.20 and later versions of VPLUS include: the RELATE command (introduced in B.03.20), the ability to remove function key labels, and form-by-form enabling and disabling of the AIDS, MODES and USER keys. These enhancements, the ability to execute Formspec in batch mode (as introduced in B.03.15), and the continuing support for a wide range of advanced terminal features should make the latest version of VPLUS an attractive incentive to update.

Personal Information Handling: HPListKeeper

By Moe Rubenzahl

The newest member of HP's Interactive Office is HPListKeeper. This tool allows a wide array of users to manage their own data without the aid of programmers.

HPListKeeper allows users to create, edit, find, sort, and print information in the form of lists. Even an inexperienced computer user can be creating and manipulating useful lists within 30 minutes, and will rarely need to consult the instruction manual. Menus and screen-labeled function keys (instead of commands), an online introduction, demonstrations, examples, and help features make the manual an accessory rather than a necessity. With the other HP office productivity tools, users can send lists via electronic mail, graph them, and use them to quickly create customized letters using HPWord.

The following is a sample of HPListKeeper capabilities:

- Information is stored in a list of up to 99999 entries (called records), organized in up to twelve categories (fields). Each record can be up to 992 characters long.
- Users can search a list for specific information or ranges of information. They can sort their lists, using up to three fields, in ascending or descending order.
- Users can print any fields desired in any order, so one list could be used for different purposes.
- HPListKeeper prints address labels.

HPListKeeper brings the flexibility and power of computerized information management into the hands of the people who need the information. This saves the time and expense of having programming experts handle every routine request, and offers another major advantage: it provides instant access to the information the user needs to make immediate, informed decisions.

Introducing Enhancements to HPToolset

By Tad Olson

Several major enhancements have been added to HPToolset to make it a stronger tool for applications development. These additional features are:

- HPToolset now supports Pascal applications development.
- HPToolset users can create and edit COBOL COPY libraries.
- The softkey structure has been redesigned to enhance HPToolset usability.
- HPToolset can generate source code from definitions in the data dictionary.

Pascal Applications Development

The HPToolset supports Pascal as well as COBOL applications development. All the functions available to the COBOL programmer are now available to Pascal programmers. These functions include full-screen editing, source version management, on line listings management, and symbolic debug. Applications programmers are now able to develop COBOL and Pascal programs using the same set of tools. For example, a COBOL program which contains a Pascal subprogram can be written and tested using HPToolset.

COBOL COPY Library Creation, Editing and Management

COBOL COPY library ("copylib") modules can now be edited using the HPToolset's full screen editor. HPToolset will automatically convert the KSAM format of copylib modules into a TSAM format when a copylib module is selected for editing. New commands and softkeys have been added to define and build a new COBOL library, to edit a copylib module, to list one or all copylib modules, to display copylib attributes and member module names, to purge a copylib module, to copy from an HPToolset TSAM file, a KSAM file, or an MPE file into a new or current copylib module. For more information on the new commands `>>SET LIB`, `>>LIBLIST`, `>>LIBEDIT`, `>>LIBPURGE`, `>>SHOW LIB`, and `>>LIBCOPY`, refer to the HPToolset Reference Manual (32350-90001). Essentially all the features and capabilities of the COBOL II COBEDIT utility are provided by HPToolset.

New Softkey Structure Improves Usability

The softkeys in HPToolset have been redesigned to improve usability. Previously, the softkeys were designed as a tree-like structure. Now the softkeys follow a loop-design where the keys can repeat. For example, suppose there is an edit key set and a debug key set. Greater control by the user is provided because keys have been added which permit the user to progress forward and backward through each set. Also, some keys have been renamed for greater clarity and repositioned for greater ease of use.

These major enhancements to HPToolset expand the programming capabilities for COBOL and Pascal programmers, and is the first step in evolving HPToolset into a multi-language applications development system.

Dictionary Code Generation

HPToolset can generate source code from definitions in the data dictionary. This feature allows HPToolset to use a file description in the data dictionary to generate COBOLII environments and data division entries; PASCAL TYPE or VAR declarations can also be generated.

Introducing VisiCalc/3000

By Susan Broom

VisiCalc* /3000 is a minicomputer-based implementation of the industry-standard electronic spreadsheet application written for personal computers by VisiCorp. HP's package has the complete features of standard VisiCalc plus many important additions, such as the ability to protect cells and the ability to set column widths individually.

VisiCalc/3000 gives you the ability to build spreadsheets on the HP3000 through standard HP terminals. It allows you to specify a spreadsheet by typing in formulas and numbers. VisiCalc/3000 remembers your specifications and lets you alter them as the need arises. As you enter numbers on your worksheet, formulas that use the numbers can be automatically calculated, showing you the effects immediately. This calculation facility makes VisiCalc/3000 a powerful planning and forecasting tool.

This tool can be used for applications, such as business/home budgeting and analysis, investment analysis, and inventory/sales analysis. In addition, many record keeping tasks are managed more efficiently, including income tax preparation, checkbook balancing, and the keeping of business registers.

Added Features

VisiCalc/3000 includes all the features available on standard VisiCalc: the insertion and deletion of rows; replication; math, business and engineering functions; logic functions; and a table function. In addition, VisiCalc/3000 has many advanced features not available in the standard VisiCalc program. These include:

- MULTIPLE WINDOWS, which allow multiple areas of the spreadsheet (windows) to be viewed simultaneously, either horizontally or vertically. The standard VisiCalc program allows a maximum of two windows, in either view. VisiCalc/3000 allows as many windows, in either view, as can fit on the screen.

- ON-LINE HELP, which assists the user by focusing on the portion of the help facility that is relevant to the command the user is working with. The package automatically takes the user back to where he or she left off.

- EXTENSIVE FORMATTING OPTIONS, which increase the clarity of even the most complicated worksheets. In addition to the standard VisiCalc variable format options, VisiCalc/3000 offers accounting symbols (DR for debit and CR for credit), currency symbols, parentheses or minus signs for negative numbers, automatic percentage signs, abbreviations for thousands and millions of dollars, and center justification.

- CELL PROTECTION, which protects those cells containing critical numbers, formulas, or labels from accidental entry or erasure.

- A HIDDEN CELL FEATURE, which hides sensitive or confidential information yet still includes the information in all spreadsheet operations.

*Copyright (C) Software Arts Products Corporation 1981, Hewlett-Packard 1983.

- VARIABLE COLUMN WIDTHS, which make maximum use of display and printer widths while making the final spreadsheet easier to read.

- A LARGER SPREADSHEET SIZE, with an addressable workspace of 254 columns by 254 rows. The standard VisiCalc has an addressable workspace of only 63 columns by 254 rows.

- FUNCTION KEYS, which are used for cell cursor movement, activation of the Help facility, and exiting the program.

VisiCalc/3000 has the ability to read in SD (Self-Describing) files from applications, such as HPListKeeper and Query/3000. It converts the SD file to a DIF** (Data Interchange Format, industry standard) file and then loads it into VisiCalc/3000. You may then add equations and additional data to your spreadsheet.

DIF or VisiCalc files can also be transferred to and from HP's new personal computer, the HP150, as well as the HP120 and HP125 personal computers. VisiCalc/3000 was designed to be completely compatible with VisiCalc/150 so that VisiCalc files can be brought back and forth using DSN/LINK, with no changes necessary to the spreadsheet. They have the same features.

Using VisiCalc's print file capability, a spreadsheet can be printed in a "DOCUMENT" format, then included in a TDP/3000 or HPWORD document.

Hardware and Software Requirements

VisiCalc/3000 is supported on any HP 3000 computer with the MPE IV (C/D.01.00 or later) operating system.

Terminals (all block mode terminals with 4Kb memory and format mode):

HP 2645A
HP 2647A/F
HP 2648A
HP 2622A
HP 2623A
HP 2624A/B
HP 2625A
HP 2627A
HP 2626A/W
HP 2628A
HP 2382A
HP 150
HP 3092
HP 3093

Printers:

HP 2563A
HP 2631A/B/G
HP 2608A
HP 2617A
HP 2619A
HP 2680A
HP 2688A
HP 2601A
HP 2602A

**DIF (TM) is a trademark of Software Arts.

Make HPWORD Work for You

By Mariann Tymn Ososkie and Pat Loomis

HPWORD is a very efficient word processing tool. In the hands of a careful and informed user, its performance can be improved still further. Here are some suggestions to help you get the most from the HPWORD subsystem.

Positioning Text

Rolling of text using the rolling, cursor, or syntactic-positioning keys in preparation for editing is greatly misused. This practice consumes large amounts of computer resources. There are better and faster ways:

- Use the "Edit at Edit Page" feature.

  When you wish to return to where you were last editing a document, use the "Edit at Edit Page" feature rather than starting at the beginning of the document and rolling to the point where you left off. To use this feature, perform the following steps:

  1. On the TASK SELECTION menu, type the document name and then press the "Display Doc Info" function key rather than "Edit Document".

  2. Now press the "Edit at Edit PG" function key. The text and cursor will be set to where they were before exiting the document.

  The one extra step on your part makes a big difference in performance to both you and other users of the system.

- Use the "GO TO PAGE" feature.

  The "GO TO PAGE" key on the numeric keypad is a fast and efficient way to position text when revising a document for which you have a printed copy. Once positioned to the top of the page you want, other methods (such as those that follow) can be used to find the revision point.

- Use the "Search and Replace" feature.

  Using the SEARCH feature is a more efficient positioning mechanism than rolling because only the required text is sent over the communications link to the terminal. To use SEARCH for positioning text, press the "Search and Replace" function key, fill in the string field on the menu, select the search options, press the "Manual Search" function key, then press the "Search Forward" or "Search Backward" function key. The cursor will be positioned at the desired match location, and text context will be established around this point. If the search fails, the text on the screen will not be moved. The search options are:

  1. "Search Forward" or "Search Backward".

  2. "Search as Word" or "Search as Character". In "word" mode, the matched string must be a whole word or series of words as defined by HPWORD. (A word is surrounded by spaces, and starts or ends a phrase, sentence, or paragraph.) In "character" mode, any matching string will qualify; e.g., "the" matches "their" in character mode.

3. "Match case" or "Ignore case". ("Case" refers to upper case or lower case.) With "match case", the search will succeed if a string is found with identical case. With "ignore case" mode, the match succeeds if the search object is found in any case.

For example, if you are looking for the string "a word is" at the beginning of a sentence, search for "A word is" as word and match case. This will avoid undesired matches.

- Fast positioning keys, which include:

1. NEXT PAGE and PREV PAGE hard keys position the cursor at the first character on the next or previous page.

2. Home UP or Home DOWN. Home UP (diagonal arrow key) positions the cursor on the first character on the current page. Home DOWN (SHIFT and diagonal arrow key) will position the cursor on the last character of the current page.

3. SHIFT and Cursor UP positions the line containing the cursor to the top of the screen.

4. SHIFT and Cursor DOWN positions the line containing the cursor to the bottom of the screen. The Cursor UP and DOWN keys are the vertical arrow keys on the keyboard.

5. SHIFT and NEXT PAGE positions the cursor at the first character of the last page of the document.

6. SHIFT and PREV PAGE positions the cursor at the first character of the first page of the document.

- Syntactic Positioning Keys.

To position within a page, use the PARA or SHIFT and PARA key to make coarse movements. Then use a combination of the LINE/SENT/CHAR keys to "zero in" on where you want to edit.

- Entering Centered of Flush Right Text.

When entering paragraphs that are to be centered or flush right, enter them normally as flush left paragraphs, then go back and reposition them using CENTER or SHIFT CENTER.

- How to Get to the End of a Paragraph in a Hurry.

There are two ways to jump to the end of a paragraph:

1. Use the PARA key to select the paragraph after the one you want, then SHIFT and CHAR to move backwards to the end of the previous paragraph.

2. Position yourself anywhere in the body of the paragraph other than at the first character. Then press PARA, which immediately places you at the end of the paragraph.

Document Format

It is more efficient to set up the document format before entering text, rather than to enter the text with default format and then perform a document format change.

Hyphenation

If you are producing draft material, turn off automatic hyphenation on the document format menu while you are in the text entry or revision mode. Turn hyphenation back on just before printing a final copy.

If you are composing online in preparation for printing a final copy, you may wish to leave hyphenation on so that what you see on the screen need not be revised before printing.

When hyphenation is on, it takes more of your time and more system resources as text is entered or revised.

Large Paragraphs

Paragraphs larger than 12 to 15 lines should be avoided. If the material is tabular in nature, and if you have pressed RETURN to end each line rather than allowing word-wrap to occur, you can split up the large table into multiple paragraphs without it appearing as such. Just press the NEW PARA key after about ten lines, then press RETURN and C^c to delete any paragraph spacing lines. The resulting paragraphs will appear contiguous on the screen and when printed.

Inserting Text

If you must insert a large amount of text within a line which contains tabs, it may be faster to enter the text as a separate paragraph near the point of insertion, then use the MOVE function to insert the text.

Exception Dictionary

If you notice that a particular word is not hyphenated correctly and is used many times, it is advisable to add the word to the hyphenation exception dictionary rather than repeatedly use the discretionary hyphen feature to correct the hyphenation.

Technical Notes on the WPSPACEF File

The file WPSPACEF is created automatically in every group in which HPWORD is run. This file, 1024 contiguous sectors, is created to reserve space for expanding documents. There are two types of document expansion, and both are automatic and invisible to the user. The price the user pays for automatic expansion is WPSPACEF . Points about WPSPACEF are:

- Allocation of additional extents.

  When there is insufficient group, account, or system disc space to allocate another extent in the HPWORD document being edited, the WPSPACEF file in the group is purged and the operation retried.

- File expansion.

  An HPWORD document is initially created with 1021 records in 32 extents for an extent size of 32 sectors. This will hold about 80 pages of information. If the document becomes larger than this, it expands automatically. The first expansion increases the record limit to 16381 records in

32 extents, for an extent size of 512 sectors. If HPWORD runs out of disc space during expansion, the WPSPACEF file is purged and the operation retried.

The WPSPACEF file is crucial to the correct operation of HPWORD. If WPSPACEF does not exist in a group and cannot be created, the user is warned in advance that HPWORD cannot guarantee that expansion can occur if a document becomes full. Text can be lost if HPWORD runs out of disc space and WPSPACEF does not exist.

Here are some useful facts to keep in mind when dealing with WPSPACEF :

- HPWORD does not permit the user to have a file in his/her group named WPSPACEF . An error message will be issued, and no documents can be edited.

- The user must not alter the characteristics of an existing WPSPACEF or its lockword. File equations for WPSPACEF are ignored.

- If editing a document in another group or account which the creator released, the WPSPACEF mechanism may not work depending on group and account security and whether WPSPACEF already exists. The user is warned if WPSPACEF does not exist and cannot be created.

- If disc space is so limited that purging WPSPACEF files makes a difference, continued operation of HPWORD is not recommended.

- Although there may be plenty of disc space, it may be highly fragmented. To detect this problem, run FREE5.PUB.SYS; refer to the MPE V System Utilities Reference Manual (32033-90008) for details. Remember that HPWORD requires 32-sector extents for "small" documents (less than 1021 records), and 512-sector extents for "large" documents.

- The presence of WPSPACEF does not guarantee that document expansion will succeed, because someone outside of HPWORD can claim the space as soon as it is made available. A global RIN serializes this operation within the community of HPWORD users.

Printing the Document Catalog to a Local or Slaved Printer

Many HPWORD users want to print the document catalog to their local daisywheel printer instead of the line printer, as it may not be easily accessible to them.

This can easily be done without even leaving HPWORD. Basically, this is done by saving the catalog to an ASCII disc file and then converting it to an HPWORD document. It can then be sent to any HPWORD-configured printer and edited just like any other document.

To capture the HPWORD document catalog for a given group in an ASCII file, perform the following:

1. Be sure the HPWORD terminal is configured to have access to a line printer. Unless this is done, the "PRINT CATALOG" key is not activated.

2. Log on to the group for which you want to generate the catalog.

3. Buiid a file as follows:

```
:BUILD catname; REC= -128,2,F,ASCII;DISC=1023,32,1
```

4. Enter a `:FILE` equation to link the catalog to this file:

   ```
   :FILE CATALOG=catname ,OLD;DEV=DISC
   ```

5. Run HPWORD and display and print the catalog.

An ASCII file containing the catalog has been created. To sort the file into author, subject, creation date, or any other field, use SORT. The ASCII file can also be converted an HPWORD document. In the DOCUMENT NAME field of HPWORD, enter the catalog name *catname*. Press OTHER KEYS and CONVERT DOCUMENT. Enter a new document name and press CONVERT. The document catalog will then appear on the screen as an HPWORD document; it can be edited, reformatted or printed to a local printer. (By default, it is 102 characters wide.)

A UDC, which converts the catalog and runs HPWORD, can easily be created, thus making it transparent to the user. For example:

```
HPWORD
CONTINUE
PURGE catname
BUILD catname; REC=-128,2,F,ASCII; DISC=1023,32,1
FILE CATALOG=catname,OLD; DEV=DISC
RUN HPWORD.PUB.SYS
**********
```

Unless the file *catname* has been built, you may wish to add logic that refuses to run HPWORD. If the file is created smaller than the number of records actually present, HPWORD will abort when PRINT CATALOG is pressed.

Some Necessary Cautions

HPWORD version A.02.06 includes many enhancements, but their limitations must be considered. The areas of concern are:

- On a heavily loaded system, data overruns can occur. HPWORD will try five times to transmit data to the host and if unsuccessful, aborts with a "`WPERR 35/FSERR 22` " software timeout error and the terminal hangs. This can occur:

  1. When six or more users are downloading HPWORD to the HPWORD terminal at the same time.

  2. When large amounts of text are being transmitted from the host computer to the terminal, and there is a heavy CPU or data communications load on the host computer.

  3. Mostly on ADCC-connected stations and significantly less on the ATP-connected stations, and when the terminal is operating at 9600 baud.

  To recover from a data overrun, turn off the terminal, `:ABORT` the HPWORD session if it's still running, and run HPWORD again. If the problem persists and is continuous, it is recommended that the baud rate be reduced.

- Printing documents that contain figure spaces (graphics) on the HP 2680A Laser Printer requires at least 10K sectors of contiguous space on the disc assigned to the spooler. When space is not available, the text prints successfully but without any graphics. Typically, your system manager must compress the disc to remedy this situation.

- Avoid creating documents larger than 600 pages. If large documents must be processed, they should be split into multiple smaller documents. Use different starting page numbers in each section to make the documents appear as one when printed. This is especially important if text is routinely included from large documents into other large documents.

- Back-referencing file equations allow access to documents on other computers. No other characteristics of a document should be changed, or an abort with loss of text can occur. You cannot print HPWORD documents with a back-referencing file equation.

- If you save a modified configuration while users are running HPWORD, the ports that are still running HPWORD remember the old configuration (they only look at the configuration when they first run HPWORD). If the user tries to print a document, the document may be sent to old print queues and be lost. Therefore, be sure that any station that is being reconfigured is not running HPWORD at the time the configuration is saved. If this occurs, exit HPWORD and run HPWORD again.

  Generally, it is OK to add new stations and printers or to modify configurations for stations that are not running. It is not OK to delete or change the ports for printers unless you are sure that all stations that have access to those printers are not running HPWORD.

- HPWORD has the capability of increasing its timer values in order to provide longer transmission times. To increase the timers you specify the HPWORD terminal as REMOTE in the HPWORD Configurator. The need to specify REMOTE is dependent on how the HPWORD terminal is connected to the system, your local datacomm devices, line quality, etc. If the HPWORD terminal is configured as LOCAL and timeouts are occuring, it is recommended that the stations be reconfigured to REMOTE.

- When configuring an HP 2601 or HP 2602 as a slave printer to the HPWORD terminal, the terminal configuration Datacomm 2 must be set to Full Duplex Hardwired only; other configurations are not supported. Parity may be set as desired, but be sure the configuration of Datacomm 2 and the printer match. If you use the default Datacomm 2 configuration (0's), set both the Parity Enable and the Even Parity switches on the printer OFF. You must also set the XmitPace to XON/XOFF in Datacomm 2 configuration.

- Column data must be separated by tabs, not spaces. Column data separated by blanks might appear aligned when displayed, but the Column functions (move, copy, delete, and total) interpret text separated by blanks as one column. One way of avoiding this problem is to use the CHAR key to scan for blanks in the text to verify that tabs are used to separate columns. Another way of avoiding this problem is to use Manual Search and Replace to scan for blanks.

  When documents with columns of data are imported to HPWORD from other text processors with the Convert Document feature, the conversion program does not use tabs to construct the columns of the tables (it uses spaces to achieve the desired appearance). If the user then attempts to use any of the column manipulation features (Move Column, Copy Column, or Delete Column) or the Column Total feature of the calculator, the results will not be correct.

  Refer to the HPWORD Reference Manual (32120-90001, E0983) for the exact definition of columns.

How to Write HPWORD Bug Reports

By Mariann Tymn Ososkie

HPWORD runs in both the HP 3000 host processor and the HPWORD terminal processor. In general, the HPWORD terminal handles keystrokes, menu, and function key operations, and is in overall control of all word processing functions. The host processor receives "commands" from the terminal processor and generally performs document-wide functions such as pagination, document format changes, and printing. The HPWORD terminal program is "downloaded" into the 128Kword terminal memory after HPWORD is run, and replaces most functions of the native mode terminal.

Send the Host Dump or Terminal Dump to HP

If the terminal software or host process software fails, a deferred dump spool file (priority 1) for device "LP" is created. Look for these dumps when either of the following conditions arise:

- If an unexpected "END OF PROGRAM" occurs, look for a spool file named WPTDUMP. This dump was produced for the terminal process, and contains register settings and other information at the time the error was detected.

- If the message "ABORT HPWORD.PUB.SYS" appears, the host detected the error and produced a spool file named WPHDUMP. This dump contains the cause of the failure, a stack marker trace, and a complete list of the stack from DL to S.

On HPWORD version A.01.03 or later, both dumps appear in the event of a terminal or host failure. You need not submit the error number, CST, or code offset after an abort because these are contained in the WPHDUMP. HP cannot respond to a Service Request without the dumps.

Please Document Your Service Requests Thoroughly

For fastest response to HPWORD problems resulting in a dump, send the following to HP:

- A copy of the dump spool file. This can be made by altering the priority of the spool file and thus printing it, or by dumping the spool file to tape using the SPOOK Utility. If there is no physical device LP on the system, alter the device for the spool file using SPOOK.

- A copy of the document is sometimes necessary. The document must be transmitted to tape using :STORE. FCOPY cannot be used because an HPWORD document has user labels. Or, submit a printed dump of the document, but use the DUMPDOC facility of HPWORD rather than an octal or hex FCOPY dump. The instructions for using DUMPDOC appear below:

  ```
  :FILE OUT; DEV=LP
  :RUN HPWORD.PUB.SYS,DUMPDOC
  Name of Document: BADDOC
  ```

  The DUMPDOC should be performed before the document goes into recovery, or prior to editing the document after an abort occurs.

  If the document contains sensitive information that cannot be released to HP, attempt to reproduce the problem on another document.

- A description of the keystrokes which lead up to the failure, if possible. For example: "On page 33 of document 'baddoc', positioned the cursor at the beginning of the second sentence of the fourth paragraph. Pressed the NEW PARA key. Moved the cursor back into the previous paragraph using the SHIFT PARA key, then deleted the paragraph using CTRL + PARA. HPWORD aborted with error 105."

- If the problem seems related to printing, send a dump of the HPWORD configuration file WORDCONF.WP.HPOFFICE. Use FCOPY, and select the ;HEX;CHAR option. Send the Logical Device Numbers (LDEV) of the devices having problems You must log on as WP.HPOFFICE to access the file.

- MPE system configuration listing.

- HP 3000 series number.

- HPWORD version number.

- Amount of system main memory.

- List of system discs, and whether they are master or slaves.

- Number, type, and baud rate of all terminals on system.

- Type of terminal connections (ATC, ADCC, ATP, etc.).

- Description of any remote processing on the system.

Undetected Errors

Sometimes, internal checking does not trap an error condition until it is too late. When this happens, the terminal may appear to be alive, but all processing suddenly stops. To recover, you must turn off power to the HPWORD terminal and redownload HPWORD again.

If the terminal process fails, the host process will detect it and produce a host dump (WPHDUMP) which will indicate "SOFTWARE TIMEOUT (FSERR 22) ". In this case, the dump itself is useless, but the keystrokes leading up to the failure are crucial. Without the keystrokes, it will do no good to submit a Service Request. It can take up to one minute for the host to detect a terminal code failure.

If the host process fails in a way that cannot be trapped (e.g. stack overflow, bounds violation), no dump will be produced. Again, the sequence of keystrokes leading to the failure is required to diagnose the problem. The fact that no dump was produced is a critical piece of information in this case.

Download Failures

Any WPHDUMP which results from a failure of HPWORD to download is crucial to solving the problem. A download failure is indicated by a hang any time from initiation of HPWORD until the TASK SELECTION MENU appears.

Be sure to include any configuration information if download failures occur. We need to know how the HPWORD terminal is connected to the system, the baud rate and parity settings, and any other HPWORD terminal configuration information available.

Release of IFS/3000 and IDS/3000

By Gary Sherwood

Version A.02.00 of the IFS/3000 and IDS/3000 software programs has now been released. This article summarizes the major enhancements and changes to this software.

Both IFS/3000 and IDSCHAR have been enhanced to recognize the existence of the new HP 2688A Laser Printer. To support this printer, many new character sets and environment files have been released. The HP 2688A is a page printer that prints at 300 dots/inch, providing excellent print quality. It prints at a maximum print speed of 12 pages/minute. It will also allow the printing of merged text and graphics.

Two new intrinsics have been added to IFS/3000, PCONVERTRASTER and PSELECTPAGE. Support of these two intrinsics has also been added to the LPS Interpreter.

PCONVERTRASTER allows the conversion of a pure dot/bit raster image into a raster format that is compatible with printing on either the HP 2680A or HP 2688A Laser Printer. The image can also be rotated in 90-degree increments, scaled in height and/or width, and inverted from white on black to black on white, or vice versa. This intrinsic provides the user with the capability to convert dot/bit raster images from any source, HP or non-HP, for printing on the HP laser printers. For more information on the PCONVERTRASTER intrinsic, refer to the article entitled "PCONVERTRASTER Intrinsic is New for Laser Printer" in this package.

PSELECTPAGE allows the user to go directly to any logical page defined in the environment, regardless of its location in the logical-page table. This intrinsic allows you to do an optional physical-page eject prior to going to the selected, logical page. Consequently, this intrinsic provides the user with much easier control of the logical page print sequence.

All menus in IFS/3000 and IDS/3000 have been updated to be friendlier to the user and more consistent with other HP3000 software. By gathering information from several long-time users of IFS/3000 and IDS/3000, we have reformatted these menus; we feel that you will be very pleased with the layouts.

The maximum ComArea size for environments used with the LPS Interpreter has been reduced from 5000 words to 3000 words. This reduction allows support of the new intrinsics.

Version A.02.00 of the IFS/3000 and IDS/3000 software is supported on MPE IV or later (C/D.00.20 thru G.00.00).

IFS/3000 Now Offers Katakana Character Sets

By Jim Stratton

Six new character sets have been added to those available with the IFS/3000 software. The new sets contain "Japanese ASCII" characters as substitution sets and Katakana characters as character extensions. The substitution character sets contain a Yen symbol in place of the US ASCII backslash symbol. The extension sets are designed to be used in conjunction (paired) with the substitution sets. The new character sets (found in CHARSETS.SYS) are:

- LNPR66K Line printer font in 4, 6, and 8 point (substitution).
- LNPR66KX Line printer font in 4, 6, and 8 point (extension).
- OCRBK Optical Character Recognition font in 11 point (substitution).
- OCRBKX Optical Character Recognition font in 11 point (extension).
- PICAK Pica font in 6, 8, and 12 point (substitution).
- PICAKX Pica font in 6, 8, and 12 point (extension).

In addition to the character sets, six new environment files use the Katakana characters. In each environment, the JASCII set is font 0, and the Katakana set is font 1. The new environments (found in ENV2680A.SYS) are:

- LPK Line printer (no reduction).
- PICAK Pica (no reduction).
- LPK2 Line printer (2:1 reduction).
- PICAK2 Pica (2:1 reduction).
- LPR4 Line printer (4:1 reduction).
- PICAK4 Pica (4:1 reduction).

The Katakana character set can be accessed either by using the eighth bit (the most significant bit of the character) or by the SHIFT IN and SHIFT OUT characters. If the eighth bit is off, the HP 2680A uses the secondary (extension) character set whenever it encounters a SHIFT IN (ASCII 14 = N^C), and the primary (substitution) character set whenever it encounters a SHIFT OUT (ASCII 15 = O^C).

If the eighth bit is on, then the HP 2680A uses the extension character set until the eighth bit is turned off. The printer then uses the substitution character set, unless a SHIFT OUT was previously in effect.

PCONVERTRASTER Intrinsic Is New for Laser Printer

By Jim Stratton

PCONVERTRASTER is a new IFS/3000 intrinsic that converts an externally formatted raster image into one that is printable on the HP 2680A Laser Printer. As a part of the conversion process, the user may choose to scale, rotate, or "invert" the raster image.

SYNTAX

```
                    LA          BA             BA
PCONVERTRASTER(comarea,infilename,outfilename,

                  BA        R      R        IA
                  device,height,width,speclist);
```

PARAMETERS

comarea

logical array, variable in length, required.
Contains information for internal use. The first word returns the status of the procedure call. (Refer to the IFS/3000 Reference Manual (36580-90001) for information on the *comarea* length and error reporting.)

infilename

byte array, 35 bytes, required.
Name of the raster image file to be converted and manipulated.

outfilename

byte array, 35 bytes, required.
Name of the raster file that will contain the manipulated results (scaled, rotated, etc.).

device

byte array, 7 bytes, required.
Name of the device for which the raster image is ultimately targeted for ("2680A").

height

real, required.
Real variable indicating the height that the raster image is to be scaled to, prior to applying the rotation. Given in the units identified in the first word of the *speclist* parameter.

width

real, required.
Real variable indicating the width that the raster image is to be scaled to, prior to applying the rotation. Given in units identified in the first word of the *speclist* parameter.

speclist

integer array, 20 words, required.
The *speclist* array defines the remaining parameters:

Word 0	Indicates the units of *height* and *width* measurement. 0=dots, 1=inches, 2=centimeters, 3=millimeters.
Word 1	Indicates the rotation of the raster image, in 90 degree increments clockwise. (0, 90, 180 or 270).
Word 2	Indicates the scaling method to be used. 0=Threshold test, 1=Pattern recognition, 2=Randomized method.
Word 3	Indicates a threshold value for use when the scaling method (Word 2) is 0 or 2. Word 3 is in the range 1 to 255. The greater the value, the lighter the scaled raster image. If set to 0, a default threshold value is calculated.
Word 4	Indicates whether the scaled raster image should be inverted (all "on" bits are turned "off", and vice-versa). If set to 1, an inverse image will be created.
Word 5 - Word 19	Not used.

All parameters are required, and passed by reference only. The requirements for the input raster image file are:

- The file must be a permanent ASCII or binary disc file with fixed length records, where each record represents a left-to-right raster scan line.

- Each record in the file is interpreted as a dot-per-bit image, where there is a one-to-one correspondence between each bit turned on (set to 1) and each dot of the printed scan line.

- Record must be of fixed length less than or equal to 124 words (1984 bits), and there can be no more than 3060 records (scan lines) in the file.

In general, the following guidelines apply to the three scaling methods. For line art graphics, use threshold testing. For graphics with text or line patterns, use pattern matching. For graphics with few defined hard lines (i.e., a picture from a newspaper) use randomized threshold testing for an "even" overall appearance. Vary the threshold value to adjust the relative gray scale of the image. Once the scaled raster image is created, it may be downloaded and printed on the Laser Printer. Refer to the IFS/3000 Reference Guide (36580-90001) for information on other Laser Printer graphics intrinsics.

HP 2680A, 2688A Error Trailer: Line-by-Line Explanation

By Rich Kaylor

Here's an explanation of the HP 2680A and HP 2688A Laser Printer error trailer. If you select an undefined character set, move the pen off the page, or cause similar formatting errors, the printer reports an error to the host, along with the number of the record in error. The host system prints the errors in an error trailer at the end of the output.

The system will abort a job after more than 100 programming errors are reported by the printer. The first 100 errors will be listed with the message "`***EXCEEDED 100 ERRORS FOR THIS JOB, SPOOLFILE DEFERRED`". The output of aborted jobs is deferred, allowing the programmer to read the error trailer and examine the bad records with the SPOOK utility.

```
:RUN SPOOK5.PUB.SYS
>TEXT devicefileid
>MODE CONTROLS = ON
>LIST firstrecnum/lastrecnum
>EXIT
```

Here is an annotated HP 2680A/2688A Laser Printer error trailer:

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         *** [1]ERROR LOG FOR LDEV = 14  [2]DFID = #0667 ***
[3]REC=1667, PHYS PAGE =1, MOVED LOGICAL PEN OFF THE LOGICAL PAGE
         ***                            ***
         ***    ENVIRONMENTAL STATISTICS    ***
         ***                            ***
[4]BUFFER = 16 BLOCKS   [5]BUCKETS AVAILABLE = 32051  [6]BUCKETS USED= 1772
[7]MEMORY=1024K WORDS
[8]3 VFC'S;  [9]1 FORMS;  [10]7 CHARACTER SETS;  [11]1 ACTIVE LOGICAL PAGES;
[12]0 PICTURES
[13]CHARACTER SETS = 43568 WORDS;  [14]FORMS = 6256 WORDS;  [15]VFCS = 192 WORDS;
[16]PICTURES = 0 WORDS
[17]PAGE LENGTH = 8.50 INCHES  (21.59 cm)
[18]PAGE WIDTH  =11.00 INCHES  (27.94 cm)
[19]FORMS CLIPPED = NO;  [20]DATA TRUNC = NO;[21]CHARS CLIPPED = 77;
[22]PICTURES CLIPPED = NO
[23]TOTAL USER MEMORY = 50016 WORDS  [24]DATE CODE = 2214
[25]TOTAL PHYSICAL PAGES PRINTED = 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

1. `ERROR LOG FOR LDEV = ` *xx* : the logical device number of the Laser Printer that produced the error trailer.

2. `DFID = #0`*nnn* : the device file identification number of the output spool file that contained the error. If the output spool file is deferred, this number is used in SPOOK to identify the spool file (see above).

3. Certain errors will cause an error message to appear. The number of the record that caused the error and the physical page where the error occurred are listed, along with an error message. If the output is deferred, you can use SPOOK to view the record(s) that have errors. The specific error messages are listed below, along with a brief explanation of corrective action:

   - NOT ENOUGH MEMORY FOR CHARACTER SET LOAD : There is not enough available memory to load the character set which is being sent to the Laser Printer:

   - NOT ENOUGH MEMORY FOR FORMS LOAD : There is insufficient memory available to load the form which is being sent to the Laser Printer.

   - NOT ENOUGH MEMORY FOR VFC LOAD : There is insufficient memory available to load the VFC which is being sent to the Laser Printer.

   - OUT OF MEMORY-JOB ABORTED,SPOOLFILE DEFERRED : The main memory is completely occupied with character sets, VFCs, forms, and data so that the current input data is lost.

   - NOT ENOUGH MEMORY FOR PICTURE DOWNLOAD : The main memory is completely occupied with character sets, VFCs, forms, and data so that the current picture cannot be downloaded.

   To correct any of the previous five errors, add additional memory to the Laser Printer, and/or reduce the amount of total user memory used (item 23). Look at the memory used for character sets (item 13), forms (item 14), VFCs (item 15), and pictures (item 16). Eliminate any of these that are not essential. Limit the range (if possible) of character sets in the character font menu of the environment by specifying the lowest and highest character codes actually used. Consider making addressable pictures temporary (item 12).

   - ATTEMPT TO SELECT AN UNDEFINED CHARACTER SET : The character set number specified is not in the environment file. The record is ignored and the currently selected character set remains active. Either add the character set to the environment or change the character set reference to a valid one.

   - AN UNDEFINED FORM WAS SELECTED : The form name referenced by the environment or an IFS/3000 procedure, or the form number referenced by an intrinsic, does not exist. Either correct the reference, download the form with an intrinsic, or remove a previous intrinsic call which deleted the form.

   - ATTEMPT TO PRINT WITH NO VFC SELECTED : An attempt was made to print data and there is no currently selected VFC. VFCs were turned off, but an attempt was made to issue a channel call. The channel call is ignored and single spacing is performed. Either specify a VFC file in the VFC selection menu of the environment or with an intrinsic, or eliminate the channel call.

   - ATTEMPT TO PRINT WITH NO LOGICAL PAGE TABLE : An attempt was made to print data and there is no currently active logical page. Be sure to have a logical page active when printing data. Either select a logical page, or make a logical page initially active in the environment.

   - MOVED LOGICAL PEN OFF THE LOGICAL PAGE : An attempt was made to move the logical pen off the currently defined limits of the logical page. Either modify the move pen command, or change the size of the logical page.

- DATA SATURATION-DATA WAS LOST : The HP 2680A/2688A could not process all of the data fast enough to transfer it to the drum. The number of characters that may appear on any scan line is a function of the horizontal width of each character, as well as the vertical height of each character. Wider and higher characters require greater processing than smaller ones, so consequently fewer can be placed on a scan line. However, the maximum number of characters (excluding blanks) and form elements that can appear on any scan line is 255. Forms and pictures will also reduce the number of characters that can be placed on one scan line. Blank scan lines between two adjacent rows of print may help solve this problem. This error can also occur if two logical pages are superimposed on each other so that the total items per scan line exceed the limits of the machine. The 255 limit may be exceeded if you underline characters or use overriding line spacing to overlay print lines.

- SPOOLER BLOCK CONTAINS FORMAT ERROR : The information in the spoolfile was not in the correct format for the HP 2680A/2688A to interpret it. This could be an invalid function code, or the size of the record or block is incorrect in some way. Use SPOOK to analyze the record in error, as explained at the end of this article.

- A MULTI-COPY FORMS ERROR HAS OCCURRED : An attempt was made to use the multi-copy forms table but the table has not been loaded for this job. Be sure the multi-copy forms box is Y on the physical page and environment default menu in the environment, or that the multi-copy form overlay table was downloaded by an intrinsic.

- EXCEEDED MAXIMUM COPIES PER PAGE : The maximum number of copies per physical page has been exceeded. The physical page menu of the environment file allows you to specify how many times you want a physical page printed before the next physical page is printed. You specified a number greater than the maximum allowed. The default maximum number of copies per physical page is 32,767. However, the maximum value can be changed with the FDEVICECONTROL intrinsic. If this error occurs, only the maximum allowed will be printed. Note that this error is unrelated to the number of copies generated by the spooler.

- EXPECTED JOB OPEN FROM SPOOLER : A command or function code was received when no job was in progress. The printer expects that the first record received is a "JOB OPEN".

- SPACING PARAMETER IS <= 0 IN LPT FOR VFC : The VFC linespacing is less than or equal to zero. The base height of the character set was non-positive. Check that word 10 (height of base character set) of the logical page table downloaded by the FDEVICECONTROL intrinsic is correct.

- ATTEMPT TO SKIP TO A NON-EXISTENT VFC CHANNEL : Change your VFC call to point to a valid VFC channel, modify your VFC file to include that channel, or reference a VFC file which has that VFC channel.

- LOGICAL PAGE WAS TRUNCATED TO FIT ON PAPER : Change the size of the logical page or physical page, or allow scaling on the logical page forms menu in the environment.

- OPERATOR MOUNTED WRONG SIZE PAPER : The page size requested by the user did not match the paper length set by the operator. The page length set by the operator will prevail. Rerun using correct size paper, have the proper page length entered on the HP 2680A/2688A operator panel, or change the physical page and environment defaults menu in the environment. You will get this message if you print on the HP 2688A with an HP 2680A environment, or vice versa.

- ATTEMPT TO PRINT WITH NO CHARSET SELECTED : No character set was selected when a print record was processed. The record skipped. Specify a valid character set in the environment, or select a valid character set with a procedure or intrinsic.

- ATTEMPT TO PRINT WITH TOO MANY PICTURES ON A PHYSICAL PAGE : Attempt to print more than 64 pictures on a physical page. This is a system constraint. Modify your program.

- ATTEMPT TO PRINT A PICTURE WHICH IS NOT PRESENT : Modify the program to reference a valid picture, or make a picture available.

4. BUFFER = *xx* BLOCKS : The size of the incoming data buffer in 512 word blocks. Rather than receiving one data record at a time, the HP 2680A/2688A expects large, multiblock transfers. Each transfer consists of one to eight 512-word blocks which were created by the host system spooler. This increases throughput and relieves the burden of deblocking from the host.

5. BUCKETS AVAILABLE = *xx* : The number of 32-word blocks of memory (buckets) available. These buckets are used in data linking and command processing. The number of buckets determines how many pages the printer can process (link up) ahead of the actual printing.

6. BUCKETS USED = *xx*. The maximum number of 32-word memory blocks used since the beginning of the job, depending on the size of the data.

7. MEMORY = *xxx* WORDS : The number of memory words in the HP 2680A/2688A Laser Printer, a minimum of 128K 16-bit words. Up to one megaword of memory is available. The mainframe and power supply can support up to one megaword. About 23K words of printer memory is allocated to printer use for system tables and cannot be accessed by the user. These system tables are invariant in size regardless of the amount of memory installed. The remainder of memory is available to the user and is called into the user area. This memory space contains all character sets, forms, VFCs and pictures which are downloaded into the user area. It also contains 32-word blocks (buckets) which are used in data linking and command processing.

8. *xx* VFC'S : The number of VFCs loaded in the printer. The printer supports up to 32 VFCs simultaneously, one for each defined logical page. A VFC consists of a two-word descriptor block, then a table of *n* words. Each bit in a word corresponds to one of the 16 VFC channels. Each word in the table corresponds to one line of the logical page. When a VFC command is received, the printer calculates the line on the page where the pen is currently located. It then accesses the VFC table at this address +1, and starts searching the VFC table looking for a word with the bit set corresponding to the VFC channel being skipped to. When the bit is found, the pen is moved down the page to the new line position, and the pen is left justified on the logical print line.

   If the VFC bit is not found before the end of the table or the end of the page, then a logical page eject occurs and the VFC table of the next logical page is scanned from the beginning looking for the bit. If no bit is found on that VFC table then the pen is placed on the first line of that VFC table.

9. FORMS = *xx* : The number of forms loaded in the printer. Electronic forms are a key feature of the HP 2680A/2688A, as they allow the use of plain paper, and the printer automatically puts a form on the page electronically. The printer allows a physical page to be divided into at most 32 logical pages. In addition each logical page may have up to two forms overlayed on it. A check is made to see if the form overflows the physical page, but no check is made to see if the form overflows the logical page. A single form, stored only once in the HP 2680A/2688A's memory, can be used by several different logical pages. The printer will relocate each use of

the form to overlay the corresponding logical page properly. The orientation of a form is dependent on the rotation of the logical page. The number of logical page form overlays on any physical page cannot exceed 30 forms.

In addition, the user can specify that two forms be overlaid on the physical page. For each copy of the same sheet of paper, the printer can be instructed to overlay two different forms on each of the first eight copies. This feature is controlled by the multi-copy forms overlay table. This means that a total of 32 forms may be placed on one sheet of paper.

The reason for two forms on each logical or physical page is to allow the user to construct forms in two parts. Generally there is a base part, common to all the parts. The second part contains the portion of the form unique to this part, such as "purchasing part" vs. "shipping part". It is allowable, however, to place two completely independent forms on a page.

10. CHARACTER SETS = *xx* WORDS: The number of character sets loaded in the printer. The maximum is 32.

11. *xx* ACTIVE LOGICAL PAGES : The number of active logical pages. The term "logical page" describes an area of data, which with rotation and VFC, may or may not occupy an entire physical page. The **HP** 2680A/2688A allows up to 32 logical pages per physical page (a physical page is one sheet of paper, bounded by perforations).

12. *xx* PICTURES : The number of addressable pictures currently loaded. The maximum is 32. A user wishing to output graphics to the HP 2680A/2688A will do so by using a construct called a picture. A picture is a dot/bit image of a print area which has been partitioned into a set of picture cells.

    Two types of pictures are provided, temporary and addressable. A temporary picture is downloaded, then printed as data and deleted. An addressable picture is downloaded, stored in memory and available for printing on demand by the user.

    These two picture types give the HP2680A/2688A the flexibility to efficiently address conflicting requirements:

    - A job which prints multiple copies of the same or different pages may use addressable pictures, as directed by the programmer or creator of the job. The job downloads the picture, assigns it an index, and references the picture by index whenever desired. Addressable pictures minimize data communication overhead when the same picture must be printed many times. However, memory limitations must be considered.

    - Jobs which print many pictures, but which rarely repeat them, use temporary pictures. Whenever the job requires the picture to be printed, it issues a print picture command which downloads the picture and prints it and releases memory after printing.

    The HP 2680A/2688A supports up to 32 addressable pictures and 64 total pictures based on memory limitations.

    Both types of pictures may be used simultaneously, within the same job, in any manner the creator wishes. The only absolute restrictions are memory availability and performance limitations of the HP2680A/2688A.

13. CHARACTER SETS = *xx* WORDS : The number of HP 2680A/2688A memory words used for character set definition.

14. FORMS = *xx* WORDS : The number of HP 2680A/2688A memory words used for form definition.

15. VFCs = *xx* WORDS : The number of HP 2680A/2688A memory words used for VFCs.

16. PICTURES = *xx* WORDS : The number of HP 2680A/2688A memory words used for pictures.

17. PAGE LENGTH = *xx* INCHES (*yy* cm) : The length of the page in the direction of motion, in .25 inch increments. The page length handled by the HP 2680A/2688A may vary from 3 to 17 inches (76.20 to 431.8 mm) in .5 inch increments. The page length of the HP 2688A is 11.8 inches (297 mm).

18. PAGE WIDTH = *xx* INCHES (*yy* cm) : The width of the page in .1 inch increments. The page width from tractor to tractor may vary from 6.5 inches (165mm) to 12.7 inches (322mm). The maximum print width is 11.38 inches. The page width on the HP 2688A is 8.5 inches (210 mm).

19. FORMS CLIPPED = (YES/NO) : If yes, a form or forms were not printed on a physical page because they started in the nonprintable area of the physical page. Either a programming error occurred or the operator used the registration buttons to locate the form off the page.

20. DATA TRUNC = (YES/NO) : If yes, data was truncated from the top or bottom of the physical page. Either a programming error occurred or the operator moved the print off the page with the registration buttons.

21. CHARS CLIPPED = *xx* : The number of non-blank characters clipped (not printed) this job. As characters are processed, the bounds of each character cell are computed to ensure that the entire character lies within the limits of the active logical page. If a character fails this check, it is skipped and no attempt is made to print it. If a character cell is not completely within a logical page, the character is thrown away and the number of characters clipped is incremented, and the actual pen position remains unchanged. There is one exception to this rule: If a character cell is orientated in the same direction as the logical page, and it lies off the bottom of the logical page, a logical page eject is performed and the character is placed in the first cell position of the next logical page.

22. PICTURES CLIPPED = (YES/NO) : If yes, a picture or pictures were not printed because they started in the nonprintable area of the page. Either a programming error or the operator used the registration buttons to locate the picture off the page.

23. TOTAL USER MEMORY = *xx* WORDS : The number of user area words actually loaded.

24. DATE CODE = *yyww* : The date code of the DCS firmware ROMs installed, in the form *yy/ww* . The *yy* shows the year since 1960, and *ww* shows the week of the year in the range 1 to 52. Date code 1905 translates: to the fifth week of 1979 (1960+19=1979).

25. TOTAL PHYSICAL PAGES PRINTED = *xx* : The number of sheets of paper printed since beginning of the job.

Graphics Support Considerations With Upgrade to MPE V/E

By Pat Gage

The new versions of HPDraw, HPEasyChart and DSG/3000 provide many new features, support several new devices, and include performance improvements.

These three graphics subsystems include the following new features:

- New fonts with a variety of type styles.
- Ability to create 35mm slides with HPDraw.
- Coordinated color palettes for plotters and 35mm slides.
- Expanded Figure Library.
- Current objects (under HPMENU) to allow information to be shared between the office products.
- User Interface standards.
- LDEV# (logical device number) fields for remote plotting.

For a detailed explanation of the features that are specific to each of these products, refer to "FEATURES BY PRODUCT" in this article.

The new devices supported by HPDraw, HPEasyChart, and DSG/3000 are:

HP 2688A Laser Printer
HP 2563A Line Printer
HP 2608S Printer
HP 7475A six-pen Plotter
HP 150 Personal Computer
HP 2625A/2628A Terminals

Make sure your System Tables are configured correctly when you run DSG/3000, HPDraw, and HPEasyChart. The configuration is included in "SYSTEM TABLE CONFIGURATION" in this article.

HP DSG/3000 Intrinsics Changes

Two new intrinsics for making scattergrams have been added to DSG/3000. GLINESET and GLINEINQ sets or inquires the scattergram option for line charts. When selected, data points are plotted using point markers without connecting lines. You must call GLINESET with SCATTER=1 for a scattergram; SCATTER=0 (default) is for a line chart.

Aside from the addition of these two new intrinsics, changes have been made to twelve DSG/3000 intrinsics. These twelve intrinsics are as follows:

GDEVICESET	GDEVICEINQ
GSUPPRESSSET	GSUPPRESSINQ
GTEXTSET	GTEXTINQ
GTEXTASET	GTEXTAINQ
GTEXTCONTROLSET	GTEXTCONTROLINQ
GFONTSET	GFONTINQ

Programmatic changes to these intrinsics will require users of DSG/3000 applications to recompile their programs if they have installed the new graphics software. The section "NEW PARAMETERS AND INTRINSICS" in this article includes the new parameters added to these intrinsics, plus the parameters for the two new intrinsics. You can also refer to Section VII, "PROCEDURES--REFERENCE SPECIFICATIONS", in the DSG/3000 Reference Manual (32250-90001).

Figure File Format Changes

Figure files created using DSG/3000 version A.01.00, HPEasyChart version A.00.00, or HPDraw versions A.00.00 and A.00.01 will need to be converted for use with the present version of HPDraw. A program provided with HPDraw will convert the old figure files to the new format. This program is FUPDATE.PUB.SYS; it requires you to be the creator of the files being converted. To use this program, type:

```
:RUN FUPDATE.PUB.SYS
```

A message is displayed on the screen instructing you to enter the name of the figure file to be converted, then to enter a unique name for saving the old figure file. When you enter the two names following the colons, you receive a message stating that the figure file was successfully converted. You can continue to enter names if you wish; the first instruction will be repeated. Press RETURN to exit the program. (For more information on Updating Figure Files, refer to "HPDraw MAINTENANCE" in the HPDraw Reference Guide (32108-90001).)

Drawing File Format Changes

The HPDraw drawing file structure has been modified to reflect the new fonts and palettes. The new version of HPDraw will automatically convert old HPDraw drawing files when they are edited and saved.

It is important to emphasize that HPDraw files created/modified with HPDraw A.02.00 (MPE V/E) cannot be accessed by an older version of HPDraw software; doing so may result in damage to the new version of the files.

Overall Version Compatibility

The following lists show the compatibility differences between the old and new versions of the three graphics subsystems:

Running the new version of programs on old version files:

DSG (A.03.00):	Use FUPDATE to convert figure files.
EZCHART (A.02.00):	Use FUPDATE to convert figure files.
HPDRAW (A.02.00):	Use FUPDATE to convert figure files.

Running the old version of programs on new version files:

DSG (A.01.00):	Not recommended. Will not work with new data types; no file damage.
EZCHART (A.00.00):	Although no action is necessary, not recommended.
HPDRAW (A.00.00):	Not recommended. Do not access new version files; doing so can damage the new version drawing files if they are edited and saved.

For DS users, you can only transfer the graphics files from a system that uses older versions of graphics software to a system that uses newer versions of the software. You cannot transfer from a new version back to an old version.

Plotting to Stand-alone Plotters

The graphics products will allow the user to plot to stand-alone plotters (a plotter connected directly to a port) without using file equations. DSG/3000 and HPDraw have been modified to allow the user to specify the logical device number on the plotting menus. (Refer to "LOGICAL DEVICE NUMBER" in the DSG/3000 Reference Manual (32250-90001), and "EXPANDED PLOT MENU" in the HPDraw Reference Guide (32108-90001).) HPEasyChart requires the use of an ;INFO= string on the :RUN command to plot to a stand-alone plotter. (Refer to "INFO STRING" in the HPEasyChart Reference Guide (32109-90001).)

When programmatically directing figures to a device during a session, do not use file equations for the GRAPHDEV and ACTIVED formal file designators; the use of these designators is not supported in the new versions of the online graphics software. (Refer to "FORMAL FILE DESIGNATORS" in the HPDraw Reference Guide (32108-90001).)

Background Jobs

This section includes a discussion on vector-to-raster conversion and producing a 35 mm DICOMED tape.

VECTOR-TO-RASTER CONVERSION. The new graphics software, by default, will make use of a background vector-to-raster conversion job when you upgrade to MPE V/E. The conversion job is invoked from the plotting menus when you specify either "2680A", "2688A", "2608S", or "2563A" as the plotting device. Note that the background job logs onto MPE as GRAPHICS.HPOFFICE,GRAPHICS for all users. Also, the background vector-to-raster job makes temporary use of "RC" files, which appear in your account during the conversion and are later purged. The files are named with "RC", followed by a time stamp. (Refer to "SETTING UP AN HPOFFICE ACCOUNT" in this article.)

PRODUCING A 35 MM DICOMED TAPE. When you specify "D48" as the plotting device, this results in a background job being run that produces a DICOMED 35mm tape. This tape can be processed into 35mm slides at a DICOMED Service Bureau. Note that the background 35mm job makes temporary use of "D" and "F" files while making a 35mm tape. These files appear in your account and are later purged. The files are named "D" and "F" and are followed by a date stamp. (Refer to Appendix C, "35MM SLIDES", in the HPDraw Reference Guide (32108-90001) for a list of DICOMED Service Bureaus.)

If you need to change these background jobs to add passwords, or change a user group or account, refer to "CONFIGURING THE GRAPHICS PRODUCTS" in this article.

Hardware Notes

The HP 150 needs to have REV B firmware to be able to use the HP 7475A plotter connected to the HP-IB port. An Upgrade Kit (45611-60017) is set up; it is available from CPC (supplier 1500) and will contain all REV B ROMs.

For full support when using the graphics products, you should make sure your HP 150 is configured as a 150 (not an HP 2623A). (Refer to Appendix A, "CONFIGURING YOUR HP 150", in the Owner's Guide (5953-5860).)

The following plotters, when connected to HP 2700 or HP 2647F terminals, are not supported by HP 3000 Graphics when connected in eavesdrop mode:

HP 2700	**HP 2647F**
HP 7221A/B/S/C/T	HP 7470A
HP 7220A/S/C/T	HP 7475A
HP 7475A	HP 7580A/B
HP 7580A/B	HP 7585A/B
HP 7585A/B	

The workaround for each subsystem is to first connect the plotter directly to a port (plotter stand-alone), then:

- For DSG/3000, type the LDEV# in the appropriate field on the Main menu. Your data will be plotted to the LDEV# you specified.

- For HPDraw, type the LDEV# in the field on the Expanded Plot menu. (You must be in Advanced User level for this field to be displayed.) Again, your data will be plotted to the LDEV# you specified.

- For HPEasyChart, use the following Info String when you enter the :RUN command:

  ```
  :RUN EZCHART.PUB.SYS;INFO="DEV=ldev#"
  ```

  (Refer to Appendix A, "INFO STRING", in the HPEasyChart Reference Guide (32109-90001).)

Features by Product

This section includes a discussion on new features/enhancements for DSG/3000, HPEASYCHART, and HPDRAW.

NEW FEATURES AND ENHANCEMENTS FOR DSG/3000. They are as follows:

- Charts may be rotated to take advantage of the paper width on plotters that scroll. On the Main menu type an "X" in the Rotate field to use this option. Rotation can be specified programmatically through the intrinsic GEXECHART. No rotation is the default specification.

- Annotations may be suppressed from the Graphing Options menu. When you type an "X" in the Annotations field under Suppress Options, the annotations are not plotted when the chart is plotted; however, they remain in the chart file. Annotations are suppressed programmatically through the GSUPPRESSSET and GSUPPRESSINQ intrinsics.

- A minimum axis value not equal to zero can be input for bar charts on the Bar-Chart Axes menu. Setting a non-zero minimum value can cause bigger distinctions between data values. The default bar-chart axis minimum remains at zero.

- You can display preset values for all menus displaying the function key "Show Defaults" (f4). If values other than the default had been entered, these values would still remain in effect.

- All data types available through QUERY/3000 and other data base systems are valid data types for DSG/3000 data files. These new types include: Integer (I), Real (R), Packed Decimal (P), Zoned Decimal (Z), COBOL Computational (J), and Binary Values (K). Only Numeric (N) and Text (T) data types can be displayed, however. These are defined on the Data Definition menu.

  QUERY/3000 and INFORM can create all these types; COBOL programs can create type J, I, Z and P; Pascal and SPL can create types I and R.

- Self describing (SD) files created through DSG/3000, QUERY/3000, HPWORD, and other HP products can be defined in the Data Definition menu by specifying the filename only. Now, rather than defaulting to the first nine variables, you can also specify which ones to plot. When you type in the variable name, the data format and type are displayed.

- You may choose not to sort the data values for line charts. Delete the "X" in the Sort field on the Line Chart menu. If you do not sort the data, it is plotted in the order in which it was entered. When the data is sorted, it is plotted in increasing values of the independent variable. The default is to sort the independent data values.

- There is a Palette field on the Graphing Options menu where you type in the name of a color palette. These palettes have predefined colors for use with color terminals or in the creation of 35mm slides with HPDraw. The palette names are as follows:

  Autoplot2700
  Basic2627
  Basic2700
  Fill/White
  Fill/Black
  Fill/Blue
  Fineline/White
  Fineline/Black
  Fineline/Blue
  Plotterpen_Fill
  Plotterpen_Fine
  Rainbow
  Shades/White
  Shades/Black
  Shades/Blue

  You may specify up to 64 pen numbers when using these palettes. White, black, or blue indicate the background color for your chart. Refer to Appendix C, "THE COLOR PALETTES", in the DSG/3000 Reference Manual (32250-90001) to see the actual colors in the palettes.

- On the Graph Options menu there is a LDEV# field for you to enter the LDEV# of a plotting device not connected to your terminal. You can reset the field by blanking out the logical device number.

- Three new fonts have been added to DSG/3000: Classic, Modern, and Uniform. The characteristics of the fonts are Boldness, Italics, and Outline. You specify the font name and characteristics on the Font Definition menu. For example, Boldness has light, medium, or bold to choose from. Italic characteristic slants the characters at an angle, and Outline indicates the characters are outlined only, not filled in.

NEW FEATURES AND ENHANCEMENTS FOR HPEASYCHART. They are as follows:

- Now you can browse the names of all the charts in the current group and account.
- "INSERT LINE" and "DELETE LINE" keys have been added to the second level of function keys on the Chart Design menu. Use these keys to add or remove entire lines of data.
- A new function key level that allows you to pass data from one chart menu to another has been added. "HOLD DATA", "RELEASE DATA", and "CLEAR DATA" keys have been added to the third level of function keys on the Chart Design menu.
- You may read a Self-Describing (SD) file produced by another office product, such as HPListKeeper, INFORM and GRAPH with the addition of the Data Selection menu. The Data Selection menu enables you to select an SD file and retrieve data by specifying the data items within the file that you want to use. Note that only data types "T" (textual) and "N" (numeric) are allowed.

  You reach the Data Selection menu by specifying Data file on the third level of function keys from the Chart Design menu.
- The maximum number of x-axis data points for the numbered line chart and scatter chart is now 60, to be consistent with the number of data points allowed on bar charts.
- The "HPIB/RS232" key has been removed from the Plot Chart menu. It is no longer needed because HPEasyChart communicates with the plotter in the appropriate protocol.

NEW FEATURES AND ENHANCEMENTS FOR HPDRAW. They are as follows:

- Two user levels now exist in HPDraw. They are Regular (default) and Advanced. Advanced level is improved so you can do the same functions as Regular, but with fewer keystrokes. You change the user level on the Main menu by typing an "A" in the User Level field and pressing (ENTER).
- Three solid fill fonts have been added to HPDraw. They are Modern, Classic and Uniform.
- Two new fields have been added to the Text menu. The Type Style field and the Outline field. The type styles are Light, Medium, Bold, Italic and Outline (Advanced User only). You can have a variety of type styles with each of the new fonts. For example, in the Type Style field you can specify "BI" (Bold Italic), "MI" (Medium Italic), "LI" (Light Italic), "BO" (Bold Outline), or "BIO" (Bold Italic Outline). The Outline Pen field is displayed for Advanced Users only.
- One new field and one new function key have been added to the Line and Arc menu. The new Area Texture field has a choice of eight area fill textures for arcs, arrows, boxes, and circles. To see the eight textures, press the "HELP" function key and type "AREA TEXTURE" in the keyword field. The textures are displayed on your screen. With the addition of the "COPY" key, you can copy objects from one section of the work area to another. If you are in Advanced User level, you can also do connecting lines and arrows on the Line and Arc menu.
- Advanced user's can specify logical device number on the Expanded Plot menu. This allows you to plot to a device not attached to your terminal.

- HPDraw offers fifteen color palettes for color graphics terminals or for 35mm slides. Color palettes are a coordinated set of colors that work well in combination. The Utility menu has a Color Palette field where you specify which color palette and which background color you want to use. (Refer to "NEW FEATURES AND ENHANCEMENTS FOR DSG/3000" in this section for a list of color palette names.) Appendix C in the HPDraw Reference Guide (32108-90001) shows the actual colors for all fifteen color palettes; it also recommends how to design your slides.

- Now you can create, edit, and save a roster file in HPDraw, instead of using other editors outside of HPDraw.

- Y^{C} has been added to HPDraw. You can use Y^{C} to stop plotting in the Plot menu.

- You can produce 35mm slides from drawings created in HPDraw and saved in a roster file. You specify "D48" on the Plot menu; this stores the slide images on a tape for processing by a DICOMED Service Bureau. The list of DICOMED Service Bureaus are in the HPDraw Reference Guide (32108-90001).

- You can browse figures or browse drawings from HPDraw instead of using the :LISTF command from MPE. The "BROWSE" function key is displayed on the Figure menu, Save menu, Plot menu, and the Edit Digitization menu.

- The Figure Library in HPDraw has been expanded to include 20 different subjects with dozens of new figures added to each figure file.

- You can edit any figure when you select Digitize on the Main menu.

- The HP 17623A Graphics Tablet can be used with an HP 2623A or HP 2627A graphics terminal for sketching, tracing, pressing functions keys, and incrementing field values.

- When you use the area fill feature with an HP2627A Color Graphics Terminal and the new color palettes, you have 64 colors to choose from.

System Table Configuration

The system resources necessary to run DSG/3000, HPDRAW and HPEASYCHART are as follows:

1. MEMORY: 1 meg strongly recommended.

2. ACCOUNTING STRUCTURE: If you are running software using DSG/3000 intrinsics that exist in a group and account other than PUB.SYS, the group must have DS, PH and MR capabilities.

   To run any of the graphics products, the user must have SF capabilities. The user's group must have file access parameter "S" set to "GU" (refer to the :NEWGROUP and :ALTGROUP commands).

3. SYSTEM TABLE CONFIGURATION:

```
MAX STACK SIZE = 31232
MAX EXTRA DATA SEGMENT SIZE = 31232
MAX # OF EXTRA DATA SEGMENTS/PROCESS = 30
CODE SEGMENT SIZE = 8193 WORDS
```

To support graphics printers, the following device classes need to be configured on the HP3000:

Device Class		Printer Model
PP	-	HP 2680A
PP88	-	HP 2688A
GLP	-	HP 2608S, HP 2563A

The following table shows how these devices should be configured in the MPE SYSDUMP Configurator dialog:

```
LOG DRT U  C T SUB                REC   OUTPUT  MODE  DRIVER   DEVICE
DEV  #  N  H Y TYPE  TERMINAL  WIDTH  DEV            NAME    CLASSES
 #      I  A P       TYPE SPEED
        T  N E

** **   0  0 32 8               66     0          S  HIOPPRT0  PP
** **   0  0 32 8               66     0          S  HIOPPRT0  PP88
** **   0  0 32 9               66     0          S  HIOCIPR0  GLP


** System Manager supplied number.
```

New Parameters and Intrinsics

New parameters were added to twelve DSG/3000 intrinsics; two new intrinsics were added as well. Any programs that use these DSG/3000 intrinsics must be recompiled. The following list shows the intrinsic specifications (new parameters are indicated by all capitals):

```
GDEVICE{SET}        (graf, device, hpib, copies, penspeed,
       {INQ}        charqual, ROTATE, LDEV, PALETTE, BACKGPEN);

                    ROTATE (integer) : 0 - don't rotate chart
                                       1 - rotate chart
                    LDEV   (integer) : Logical device to plot to
                    PALETTE (byte
                             array 16) :Color palette to use.
                    BACKGPEN (integer) :Background pen number.

GSUPRESS{SET}       (graf, chart, legends, labels, axes, title,
        {INQ}       frame, advance, ANNOTATIONS);

                    ANNOTATIONS (integer) : 0 - plot annotations
                                            1 - suppress anots.

GTEXT[A]{SET}       (graf, annotnum, x1, y1, lorg, angle, color,
        {INQ}       charsize, text, length, charfont, FILLCOLOR);

                    FILLCOLOR (integer) : Pen number for interior
                                          of text.
```

```
GTEXTCONTROL{SET}   (graf, option, color, size, font,
            {INQ}   FILLCOLOR);


                    FILLCOLOR (integer) : Pen number for
                                          interior of text.

GFONT{SET}          (graf, fontnumber, fontfile, language, SLANT,
     {INQ}          BOLDNESS, OUTLINE, charform);

                    SLANT (integer) : 0 - text without slant
                                          (default).
                                      1 - text slanted 20 deg
                                          (italic).
                    BOLDNESS (integer) : 1 - light face (default)
                                         2 - medium face.
                                         3 - bold face text.
                    OUTLINE (integer) : 0 - Outline and fill text
                                            (default).
                                        1 - Only outline text.

GLINE{SET}          (GRAF, SCATTER);
     {INQ}
                    GRAF (integer array) : DSG/3000 global info.
                    SCATTER (integer)    : 0 - line chart.
                                           1 - scattergram.
```

Setting Up an HPOFFICE Account

If you are plotting on system printers or making 35mm slides, you must set up the GRAPHICS user and group in the HPOFFICE account. If the user/group have not been created, the instructions are as follows:

```
:HELLO MGR.HPOFFICE
:NEWGROUP GRAPHICS
:ALTGROUP GRAPHICS;ACCESS=(R,X:ANY;A,L,W,S:GU);CAP=IA,BA,PH,DS,MR
:NEWUSER GRAPHICS;HOME=GRAPHICS
:ALTUSER GRAPHICS;CAP=IA,BA,ND,PH,DS,MR,SF;HOME=GRAPHICS
:BYE
```

Configuring the Graphics Products

This section includes a discussion on configuring DSG/3000 and HPEASYCHART, and on configuring HPDraw.

DSG/3000 AND HPEASYCHART. To increase performance when plotting to the 2680A, 2688A, and 2608S printers, a background vector-to-raster conversion program is streamed. This frees your terminal while the actual conversion is being performed; the plotting time will be comparable to that of any other device.

When factory released, these jobs will initially log on as GRAPHICS.HPOFFICE,GRAPHICS without any passwords. Therefore, the outputs (job listings, drawings, and charts) will be coming out under this account.

If you are satisfied with GRAPHICS.HPOFFICE,GRAPHICS, you do not need to continue reading. If you want to change the vector-to-raster conversion program to run on line instead of a background job, follow these instructions:

1. Log on as follows:

   ```
   :HELLO MGR.HPOFFICE,HP32250
   ```

2. Edit source message catalog SM1S250A.

3. List lines 1206 through 1209. Read the instruction and modify line number 1209 from "1" to "0" (1 = background job, 0 = online).

   Check by listing line number 1209. It should read as follows:

   ```
   1209     9605 0
   ```

4. If you want to add passwords or change the user, group, or account that background jobs will log on to, the instructions are as follows:

   List lines 1200 through 1205 and read the instructions. (If you have not logged on, perform steps 1 and 2.)

   For example, if you want a background job to log on to the user ARTISTS with the user password ENJOY, the group DRAWINGS with no password, and the account GRAPHICS with the account password EZCHART, line numbers 1203 through 1205 would look like this:

   ```
   1203     $    !User    !User    !Group   !Group   !Account!Account
   1204     $    !        !Passwrd!         !Passwrd!         !Passwrd
   1205     9600ARTISTS   ENJOY    DRAWINGS          GRAPHICSEZCHART
   ```

5. Now you need to create a new message catalog (which is shared by both HPEASYCHART and DSG/3000) and copy it to PUB.SYS. The instructions are as follows:

   ```
   :STREAM JMCJ250A   ** Check the job listing **
   :RELEASE P01P250A
   :BYE

   :HELLO MANAGER.SYS,PUB
   :PURGE C01C250A.PUB.SYS
   :FCOPY FROM=P01P250A.HP32250.HPOFFICE;TO=C01C250A.PUB.SYS;NEW
   :BYE
   ```

HPDRAW. To increase performance when plotting to the 2680A, 2688A, and 2608S printers, a background vector-to-raster conversion program is streamed. This frees the user's terminal while the actual conversion is being performed; the plotting time will be comparable to that of any other device.

A background job is also streamed to make a DICOMED tape when plotting the device "D48".

When factory released, these jobs will log on as GRAPHICS.HPOFFICE,GRAPHICS without any passwords. Therefore, the outputs (job listings, drawings, and charts) will be coming out under this account.

If you are satisfied with GRAPHICS.HPOFFICE,GRAPHICS, you do not need to continue reading. If you want to change the vector-to-raster conversion program to run on line instead of a background job, the instructions are as follows:

1. Log on as follows:

   ```
   :HELLO MGR.HPOFFICE,HP32108
   ```

2. Edit source message catalog SMCS108A.

3. List lines 4199 through 4210. Read the instruction and modify line number 4210 from "1" to "0" (1 = background job; 0 = online).

   Check by listing line number 4210. It should read as follows:

   ```
   4210      500 0                                   X
   ```

   This only effects the vector-to-raster conversion program performance. The background job to make the DICOMED tape cannot be customized to run on line.

4. If you want to add passwords or change the user, group, or account that the background job will log on to, the instructions are as follows:

   List lines 4186 through 4197. Read the instruction and modify the information.

   For example, if you want the background job to log on to the user ARTISTS with the user password ENJOY, the group DRAWINGS with no password, and the account GRAPHICS with the account password HPDRAW, the line numbers 4194 through 4197 would read as follows:

   ```
   4194      $   123456789012345678901234567890123456
   4195      450 ARTISTS   ENJOY                     X
   4196      451 DRAWINGS                            X
   4197      452 GRAPHICS  HPDRAW                    X
   ```

5. Now you need to create a new HPDRAW message catalog and copy it to PUB.SYS. The instructions are as follows:

   ```
   :STREAM JMCJ108A    ** Check the job listing **
   :RELEASE P01P108A
   :BYE

   :HELLO MANAGER.SYS,PUB
   :PURGE C01C108A.PUB.SYS
   :FCOPY FROM=P01P108A.HP32108.HPOFFICE;TO=C01C108A.PUB.SYS;NEW
   :BYE
   ```

READER COMMENT SHEET

Communicator 3000

5955-1770 G.00.00

We welcome your evaluation of this manual. It is one of several that serve as a reference source for HP 3000 Computer Systems. Your comments and suggestions help us to improve our publications and will be reviewed by appropriate technical personnel. HP may make any use of the submitted suggestions and comments without obligation.

Is this manual technically accurate?	**Yes [] No []**	(If no, explain under Comments, below.)
Are the concepts and wording easy to understand?	**Yes [] No []**	(If no, explain under Comments, below.)
Is the format of this manual convenient in size, arrangement and readability?	**Yes [] No []**	(If no, explain or suggest improvements under Comments, below.)

Comments:

We appreciate your comments and suggestions. This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.

Date: ______________________

FROM:

Name ______________________

Company ______________________

Address ______________________

____

____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070 CUPERTINO, CALIFORNIA

POSTAGE WILL BE PAID BY ADDRESSEE

Software Documentation Manager
Hewlett-Packard Company
Computer Systems Division/47U/MS-91
19447 Pruneridge Avenue
Cupertino, California 95014

FOLD FOLD

Part No. 5955-1770
Printed in U.S.A. 7/84
Volume 2, Issue 1

HEWLETT
PACKARD